Theses and Dissertations   1. Thesis and Dissertation Collection, all items

2009

Impacts of sigma coordinates on the Euler and Navier-Stokes equations using continuous Galerkin methods

Gibbons, Sean L.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/4856

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

IMPACTS OF SIGMA COORDINATES ON THE EULER AND NAVIER-STOKES EQUATIONS USING CONTINUOUS GALERKIN METHODS

by

Sean L. Gibbons

March 2009

| Thesis Co-Advisors: | Francis X. Giraldo |
| | Tony Eckel |

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE		*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 2009	3. REPORT TYPE AND DATES COVERED Master's Thesis, Sep 2007 – Mar 2009	
4. TITLE AND SUBTITLE Impacts of Sigma Coordinates on the Euler and Navier-Stokes Equations using Continuous Galerkin Methods		5. FUNDING NUMBERS	
6. AUTHOR(S) Sean L. Gibbons			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

In this thesis, the impacts of transforming the coordinate system of an existing x-z mesoscale model to x-sigma_z are analyzed and discussed as they were observed in three test cases. The three test cases analyzed are: A rising thermal bubble, a linear hydrostatic mountain, and a linear nonhydrostatic mountain. The methods are outlined for the transformation of two sets (set 1, the non-conservative form using Exner pressure, momentum, and potential temperature; and set 2, the non-conservative form using density, momentum, and potential temperature) of the x-z Navier-Stokes equations to x-sigma_z and their spatial (Continuous Galerkin) and temporal (Runge-Kutta 35) discretization methods are shown in detail. For all three test cases evaluated, the x-sigma_z models performed worse than their x-z counterparts, yielding higher RMS errors, which were observed predominantly in intensity values and not in placement of steady state features. Since the models did converge to a fairly representative steady-state solution the results found by this project are promising, even though they did indicate that x-sigma_z coordinates are not as accurate or efficient as x-z coordinates. With further fine-tuning of the model environment, these issues could be made minimal enough to warrant their utility with semi-implicit methods.

14. SUBJECT TERMS: Sigma coordinates, Continuous Galerkin, Euler Equations, Navier-Stokes Equations, linear hydrostatic mountain, linear non-hydrostatic mountain, explicit time integration, RK35, Runge-Kutta	15. NUMBER OF PAGES 93
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU

THIS PAGE INTENTIONALLY LEFT BLANK

IMPACTS OF SIGMA COORDINATES ON THE EULER AND NAVIER-STOKES EQUATIONS USING CONTINUOUS GALERKIN METHODS

Sean L. Gibbons
Captain, United States Air Force
B.S., United States Air Force Academy, 2003

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN METEOROLOGY
MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
March 2009

Author: Sean L. Gibbons

Approved by: Francis X. Giraldo
                            Co-Advisor

                            Maj. Tony Eckel
                            Co-Advisor

                            Philip Durkee, Chairman
                            Department of Meteorology

                            Carlos Borges, Chairman
                            Department of Applied Mathematics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In this thesis, the impacts of transforming the coordinate system of an existing x-z mesoscale model to x-σ_z are analyzed and discussed as they were observed in three test cases. The three test cases analyzed are: A rising thermal bubble, a linear hydrostatic mountain, and a linear nonhydrostatic mountain. The methods are outlined for the transformation of two sets (set 1, the non-conservative form using Exner pressure, momentum, and potential temperature; and set 2, the non-conservative form using density, momentum, and potential temperature) of the x-z Navier-Stokes equations to x-σ_z and their spatial (Continuous Galerkin) and temporal (Runge-Kutta 35) discretization methods are shown in detail. For all three test cases evaluated, the x-σ_z models performed worse than their x-z counterparts, yielding higher RMS errors, which were observed predominantly in intensity values and not in placement of steady state features. Since the models did converge to a fairly representative steady-state solution the results found by this project are promising, even though they did indicate that x-σ_z coordinates are not as accurate or efficient as x-z coordinates. With further fine-tuning of the model environment, these issues could be made minimal enough to warrant their utility with semi-implicit methods.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Numerical Weather Prediction (NWP) models are the work horse of modern atmospheric condition forecasting and as such have been the focus of numerous studies, aiming to develop more accurate models with longer deterministic forecast periods. There are countless areas in which to focus this research (spatial and temporal discretization methods, non-reflecting boundary conditions, data assimilation, prognostic equations, physical parameterizations, etc....), and for this project the focus will be on the transformation of the prognostic or governing equations from x-z coordinates to terrain following x-σ_z coordinates using a specific spatial (continuous Galerkin) and temporal (Runge-Kutta 35) discretization method. This thesis is necessary in order for future research to be able to evaluate and compare various coordinates systems while varying the temporal discretization methods, allowing for larger time steps while maintaining stability (i.e., semi-implicit methods).

Various mature mesoscale models such the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS) [2] and the Weather Research and Forecasting (WRF) modeling system [3] use a variation of the x-σ coordinate transformation. By studying the works of Gal-Chen and Somerville [4] and analyzing the transformation methods used in COAMPS and WRF, a similar x-σ_z coordinate transformation was applied to the governing equations of interest employing continuous Galerkin techniques.

The governing equations selected for this thesis consist of two forms of the Navier-Stokes equations. The Navier-Stokes equations, along with their variations, form the most widely used and accepted sets of equations for numerically resolving atmospheric flow. The first specific formulations of the equation sets selected (set 1) was the non-conservative form using Exner pressure, momentum, and potential temperature, which is used in the operational NWP model COAMPS [2]. The second formulation chosen (set 2) was the non-conservative form using density, momentum,

and potential temperature. The operational NWP model WRF [3] also uses set 2, but in a conservative form. Building on the work of Giraldo [1] who by implementing continuous Galerkin techniques, developed a 2-D (x-z slice) mesoscale model using Non-Hydrostatic Equations (Euler and Navier-Stokes Equations), the original construct was transformed from x-z coordinates to x-σ_z coordinates to test the impacts on resolving atmospheric motion in a continuous Galerkin (CG) framework. Currently, most operational non-hydrostatic models use finite difference (FD) methods (i.e., structured grids), which then rely on x-σ in order to resolve atmospheric flow in the presence of terrain. CG methods, on the other hand, can use various types of grids (unstructured grids, x-z grids, x-σ_z grids, etc....). For this reason, the CG method is well positioned to judge the effects of the coordinate system on the solution accuracy and efficiency, which is the goal of this thesis.

After the modifications were made to the model, three test cases were run: rising thermal bubble, linear hydrostatic mountain, and linear non-hydrostatic mountain. The numerical solutions were either evaluated against other model solutions (case 1) or the analytic approximations (case 2 and case 3) using root mean squared error and normalized momentum flux. The resultant data was also compared to the unmodified solutions. Additionally, the initial conditions for the test cases were pre-defined for each case, maintaining uniform initial conditions from which both coordinate systems numerical solutions can be compared.

With x-σ_z coordinates that are proven to function properly using fully explicit time integration, future research will be able to evaluate x-σ_z coordinates using semi-implicit methods. NWP models are already taking advantage of semi-implicit time integration methods, which optimize the horizontal and vertical resolutions and their associated time sets while maintaining stability. The 2-D semi-implicit method (x-z formulation) can use very large time-steps sizes since the Courant-Friedrichs-Lewy (CFL) condition is no longer constrained by acoustic and gravity waves; the penalty is that a global 2-D implicit problem must be solved. In contrast, the 1-D semi-implicit

x-σ_z formulation can use large time-steps, but it must adhere to the CFL condition due to gravity and acoustic waves in the horizontal; the advantage is that only a 1-D matrix problem needs to be solved. A model using x-σ_z can exploit semi-implicit methods along the vertical (σ) direction, which is the inspiration of this thesis topic. Both COAMPS and WRF are currently using semi-implicit methods (only along σ), but are constrained by FD spatial discretization methods.

Though only explicit time integration was used in this thesis, it is now possible to observe if the implementation of x-σ_z coordinates significantly improves or diminishes the solution of the Navier-Stokes equations over x-z coordinates when using a CG framework. Giraldo [5] has already developed semi-implicit methods for the 2-D model in the x-z framework and tested their impacts. With the model coordinates transformed to x-σ_z, future research will be able to extent the time integration to semi-implicit methods for x-σ_z and compare the results with the semi-implicit results derived using x-z coordinates. The relevant governing equations for this project, the coordinate transformation theory, and the discretization methods are discussed in Chapter II. The application of the coordinate transforms are discussed in Chapter III. The three test cases are explained and outlined in Chapter IV. A discussion and interpretation of the resulting impacts from the transformation is in Chapter V. The conclusions found and recommendations are presented in Chapter VI.

THIS PAGE INTENTIONALLY LEFT BLANK

II.   BACKGROUND

A.   GOVERNING EQUATIONS

1.   Equation Set 1: Navier-Stokes Equations with Exner Pressure

The first set of equations chosen was the Navier-Stokes equations that uses Exner pressure, of which there has been extensive amounts of documented work (i.e. this is the formulation used in COAMPS) for comparison. This set can only be written in non-conservative form and consists of a system of three equations. The first equation is the pressure tendency equation:

$$\frac{\partial \pi}{\partial t} + \vec{u} \cdot \nabla \pi + \frac{R}{c_v}\pi \nabla \cdot \vec{u} = 0 \tag{2.1}$$

$$\pi = \left(\frac{P}{P_o}\right)^{\frac{R}{c_p}}$$

where π is Exner pressure, $\vec{u}$ represents the velocity field (u, w), R is the gas constant, c_v is the specific heat for constant volume, c_p is the specific heat for constant pressure, P is pressure, and P_0 is pressure at the surface. The second equation is the momentum equation:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + c_p \theta \nabla \pi = -g\vec{k} + \mu \nabla^2 \vec{u} \tag{2.2}$$

where θ is potential temperature, g is the gravitational constant, $\vec{k}$ is a vector $(0, 1)^T$, and μ is the dynamic viscosity. The third equation is the thermodynamic energy equation:

$$\frac{\partial \theta}{\partial t} + \vec{u} \cdot \nabla \theta = \mu \nabla^2 \theta \tag{2.3}$$

For the scope of this project, only inviscid flow will be considered (i.e. $\mu = 0$) and the equations further reduce to the Euler equations which will be used in the following sections.

2. Equation Set 2: Navier-Stokes Equations with Density

The second set of equations chosen was a version of the Navier-Stokes equations that is now used in contemporary NWP models (i.e. this is the formulation used by WRF) and uses density, momentum, and potential temperature as the primary state variables. Unlike set 1, this set can be written in both conservative and non-conservative form. Though neither form can conserve energy, using the non-conservative form can still conserve mass and more sophisticated time integration strategies can be used [6]. Thus for this thesis the non-conservative form will be used, which consists of a system of three equations. The first equation is the mass equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \tag{2.4}$$

where ρ is density. The second equation is the momentum equation:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla P = -g\vec{k} + \mu \nabla^2 \vec{u} \tag{2.5}$$

$$P = P_0 \left(\frac{\rho R \theta}{P_0} \right)^{\gamma}$$

6

where P is pressure, P_0 is pressure at the surface, and γ is $\frac{c_p}{c_v}$. The third equation is the thermodynamic energy equation:

$$\frac{\partial \theta}{\partial t} + \vec{u} \cdot \nabla \theta = \mu \nabla^2 \theta \tag{2.6}$$

Similar to set 1, only inviscid flow will be considered (i.e. $\mu = 0$) and the equations further reduce to the Euler equations which will be used in the following sections.

B.   X-Z TO X-σ_Z COORDINATE SYSTEM TRANSFORM

1.   Gal-Chen and Somerville

In 1975, Gal-Chen and Somerville took the anelastic approximation of the Navier-Stokes Equation (in the cartesian form) and transformed the coordinated system to sigma-z coordinates [4]. An expanded set of prognostic equations was used for Gal-Chen and Somerville's derivation and only the first three equations and the resulting transform will be used for comparison in this project. The first equation was the continuity equation:

$$(\rho_0 u^j)_{,j} = 0. \tag{2.7}$$

where ρ_0 is density, u is the velocity components $(u, v, w)^T$, and $_j$ is the derivative operator where:

$$(\rho_0 u^j)_{,j} \equiv \frac{\partial}{\partial x^j}(\rho_0 u^j)$$

The second equation was the momentum equation:

$$\left(\frac{\partial}{\partial t}\right)(\rho_0 u^i) + (\rho_0 u^i u^j)_{,j} = -(\delta^{ij} p')_{,j} + \delta^{i3}\rho' g + \tau^{ij}_{,j} . \qquad (2.8)$$

where δ^{ij} is the Kronecker delta, ρ' is the density perturbation, p' is the pressure perturbation, and τ^{ij} is the eddy viscosity. The third equation used was the thermodynamic energy equation:

$$\left(\frac{\delta}{\delta t}\right)(\rho_0 \theta') + (\rho_0 \theta' u^j)_{,j} = H^j_{,j} . \qquad (2.9)$$

where θ' is the perturbation in potential temperature and H^j is the eddy diffusion.

Using equations 2.7 - 2.9, Gal-Chen and Somerville derived the following set of transformations [4]:

$$\bar{x} = x, \quad \bar{y} = y, \quad \bar{z} = \frac{H(z - z_s)}{(H - z_s)}$$

$$\frac{\partial \bar{z}}{\partial x} = \frac{\partial z_s}{\partial x}\frac{\bar{z} - H}{H - z_s}, \quad \frac{\partial \bar{z}}{\partial y} = \frac{\partial z_s}{\partial y}\frac{\bar{z} - H}{H - z_s}, \quad \frac{\partial \bar{z}}{\partial z} = \frac{H}{H - z_s}$$

$$\begin{pmatrix} \bar{u} \\ \bar{v} \\ \bar{w} \end{pmatrix} = \begin{pmatrix} 1, & 0, & 0 \\ 0, & 1, & 0 \\ \frac{\partial z_s}{\partial x}\frac{\bar{z}-H}{H-z_s}, & \frac{\partial z_s}{\partial y}\frac{\bar{z}-H}{H-z_s}, & \frac{\bar{z}-H}{H-z_s} \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

where the variables with the ¯ represent the variables that have been transformed, H is the height at the top of model space, and z_s is the height at the surface of the model.

8
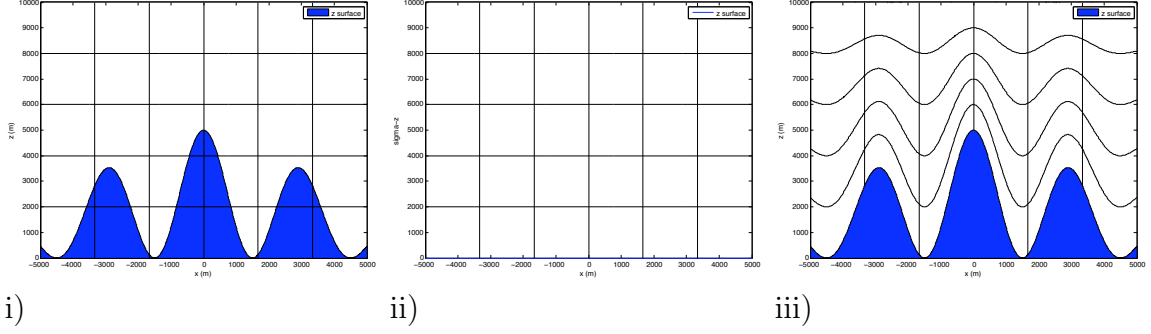
Figure 1. x-z to x-σ_z coordinate transformation: (i) traditional x-z coordinates, (ii) x-σ_z coordinates, and (iii) x-σ_z coordinates mapped back to x-z space.

These transformation functions are used to convert traditional x-z (see figure 1.i) to x-σ_z coordinates (see figure 1.ii), which when mapped back to x-z space are terrain following (see figure 1.iii). The derived inverse transformations are:

$$x = \bar{x}, \quad y = \bar{y}, \quad z = \left[\frac{\bar{z}(H - z_s)}{H}\right] + z_s$$

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} 1, & 0, & 0 \\ 0, & 1, & 0 \\ -\frac{\partial z_s}{\partial x}\frac{\bar{z}-H}{H}, & -\frac{\partial z_s}{\partial y}\frac{\bar{z}-H}{H}, & \frac{H-z_s}{H} \end{pmatrix} \begin{pmatrix} \bar{u} \\ \bar{v} \\ \bar{w} \end{pmatrix}$$

2. Basic Transformation Machinery

This section will outline the basic equations used to transform the x-z coordinates of the Navier-Stokes Equations to x-σ_z, which are similar to the transformations derived by Gal-Chen and Somerville in the previous section, but in two dimensions. The first concept used was the total differential:

$$d\tilde{x} = \frac{\partial \tilde{x}}{\partial x}dx + \frac{\partial \tilde{x}}{\partial z}dz$$

9

$$d\sigma_z = \frac{\partial \sigma_z}{\partial x}dx + \frac{\partial \sigma_z}{\partial z}dz$$

where the ~ notation indicates the transformed variable. The two total derivatives can then be written as a system of equations:

$$\begin{pmatrix} d\tilde{x} \\ d\sigma_z \end{pmatrix} = \begin{pmatrix} \frac{\partial \tilde{x}}{\partial x} & \frac{\partial \tilde{x}}{\partial z} \\ \frac{\partial \sigma_z}{\partial x} & \frac{\partial \sigma_z}{\partial z} \end{pmatrix} \begin{pmatrix} dx \\ dz \end{pmatrix} \tag{2.10}$$

Using vector notation ($\vec{}$) the above system can be simplified to:

$$d\tilde{\vec{x}} = J d\vec{x} \tag{2.11}$$

where J is the Jacobian of the transformation. Considering that the velocity $\vec{u}$ is defined by $\frac{d\vec{x}}{dt}$, the transformation can then be manipulated to become:

$$\tilde{\vec{u}} = J\vec{u} \tag{2.12}$$

Using basic linear algebra and assuming that J is invertible, which is true for an affine (one-to-one) mapping, the inverse transform for velocity can be written as:

$$\vec{u} = J^{-1}\tilde{\vec{u}} \tag{2.13}$$

The next major mathematical concept derived for the transformation was the gradient operator (∇) which, as a system of equations, can be written as:

10

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial x} + \frac{\partial}{\partial \sigma_z} \frac{\partial \sigma_z}{\partial x}$$

$$\frac{\partial}{\partial z} = \frac{\partial}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial z} + \frac{\partial}{\partial \sigma_z} \frac{\partial \sigma_z}{\partial z}$$

which when put into matrix form becomes:

$$\begin{pmatrix} \dfrac{\partial}{\partial x} \\[2ex] \dfrac{\partial}{\partial z} \end{pmatrix} = \begin{pmatrix} \dfrac{\partial \tilde{x}}{\partial x} & \dfrac{\partial \sigma_z}{\partial x} \\[2ex] \dfrac{\partial \tilde{x}}{\partial z} & \dfrac{\partial \sigma_z}{\partial z} \end{pmatrix} \begin{pmatrix} \dfrac{\partial}{\partial \tilde{x}} \\[2ex] \dfrac{\partial}{\partial \sigma_z} \end{pmatrix}$$

Similar to the total derivative, the gradient, using vector notation, was the defined as:

$$\nabla = J^T \tilde{\nabla} \tag{2.14}$$

The last major concept used for the coordinate transformation is the linear algebra identity:

$$(AB)^T = B^T A^T \tag{2.15}$$

where $A \in \mathcal{R}^{m \times n}$ and $B \in \mathcal{R}^{n \times l}$. The matrices were multiplied together and then transposed and also rearranged, transposed, and then multiplied to show equality:

$$(AB)^T = \left[\begin{pmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{pmatrix} \begin{pmatrix} b_{1,1} & \dots & b_{1,l} \\ \vdots & \ddots & \vdots \\ b_{n,1} & \dots & b_{n,l} \end{pmatrix} \right]^T$$

11

$$
= \begin{bmatrix} a_{1,1}b_{1,1} + \dots + a_{1,n}b_{n,1} & \dots & a_{1,1}b_{1,l} + \dots + a_{1,n}b_{n,l} \\ \vdots & \ddots & \vdots \\ a_{m,1}b_{1,1} + \dots + a_{m,n}b_{n,1} & \dots & a_{m,1}b_{1,l} + \dots + a_{m,n}b_{n,l} \end{bmatrix}^T
$$

$$
= \begin{bmatrix} a_{1,1}b_{1,1} + \dots + a_{1,n}b_{n,1} & \dots & a_{m,1}b_{1,1} + \dots + a_{m,n}b_{n,1} \\ \vdots & \ddots & \vdots \\ a_{1,1}b_{1,l} + \dots + a_{1,n}b_{n,l} & \dots & a_{m,1}b_{1,l} + \dots + a_{m,n}b_{n,l} \end{bmatrix}
$$

$$
B^T A^T = \begin{pmatrix} b_{1,1} & \dots & b_{n,1} \\ \vdots & \ddots & \vdots \\ b_{1,l} & \dots & b_{n,l} \end{pmatrix} \begin{pmatrix} a_{1,1} & \dots & a_{m,1} \\ \vdots & \ddots & \vdots \\ a_{1,n} & \dots & a_{m,n} \end{pmatrix}
$$

$$
= \begin{bmatrix} a_{1,1}b_{1,1} + \dots + a_{1,n}b_{n,1} & \dots & a_{m,1}b_{1,1} + \dots + a_{m,n}b_{n,1} \\ \vdots & \ddots & \vdots \\ a_{1,1}b_{1,l} + \dots + a_{1,n}b_{n,l} & \dots & a_{m,1}b_{1,l} + \dots + a_{m,n}b_{n,l} \end{bmatrix}
$$

3.   Transformation Functions

For the transformations from x-z coordinates to x-σ_z, the $\bar{x}$ and $\bar{z}$ coordinates from Gal-Chen and Somerville [4] were rewritten using new notation which will be seen through the remainder of the thesis:

$$
\tilde{x} = x, \quad \sigma_z = \frac{H(z - z_s)}{(H - z_s)} \tag{2.16}
$$

where $\bar{x}$ is now $\tilde{x}$ and $\bar{z}$ is σ_z. Using Eqs. (2.16) their derivatives were evaluated, yielding:

$$
\frac{\partial \sigma_z}{\partial x} = \frac{\partial z_s}{\partial x} \frac{\sigma_z - H}{H - z_s}, \quad \frac{\partial \sigma_z}{\partial z} = \frac{H}{H - z_s} \tag{2.17}
$$

12

which matched the derivatives found by Gal-Chen and Somerville [4]. The derivatives were then applied to various forms of the Jacobian from Eq. (2.10), which yielded:

$$
J = \begin{pmatrix} 1, & 0 \\ \frac{\partial z_s}{\partial x} \frac{\sigma_z - H}{H - z_s}, & \frac{H}{H - z_s} \end{pmatrix}
\tag{2.18}
$$

$$
J^T = \begin{pmatrix} 1, & \frac{\partial z_s}{\partial x} \frac{\sigma_z - H}{H - z_s} \\ 0, & \frac{H}{H - z_s} \end{pmatrix}
\tag{2.19}
$$

$$
J^{-1} = \begin{pmatrix} 1, & 0 \\ -\frac{\partial z_s}{\partial x} \frac{\sigma_z - H}{H}, & \frac{H - z_s}{H} \end{pmatrix}
\tag{2.20}
$$

$$
(J^{-1})^T = \begin{pmatrix} 1, & -\frac{\partial z_s}{\partial x} \frac{\sigma_z - H}{H} \\ 0, & \frac{H - z_s}{H} \end{pmatrix}
\tag{2.21}
$$

Applying Eq. (2.18) to Eq. (2.12) gives the transformation functions for the velocity field:

$$
\begin{pmatrix} \tilde{u} \\ \tilde{w} \end{pmatrix} = \begin{pmatrix} 1, & 0 \\ \frac{\partial z_s}{\partial x} \frac{\sigma_z - H}{H - z_s}, & \frac{H}{H - z_s} \end{pmatrix} \begin{pmatrix} u \\ w \end{pmatrix}
\tag{2.22}
$$

Using Eq. (2.20) and some algebraic manipulation the inverse transformation functions can be constructed:

$$
x = \tilde{x}, \quad y = \tilde{y}, \quad z = \left[\frac{\sigma_z (H - z_s)}{H}\right] + z_s
\tag{2.23}
$$

$$
\begin{pmatrix} u \\ w \end{pmatrix} = \begin{pmatrix} 1, & 0 \\ -\frac{\partial z_s}{\partial x} \frac{\sigma_z - H}{H}, & \frac{H - z_s}{H} \end{pmatrix} \begin{pmatrix} \tilde{u} \\ \tilde{w} \end{pmatrix}
\tag{2.24}
$$

13

C. SPATIAL DISCRETIZATION: CONTINUOUS GALERKIN

Continuous Galerkin (CG) methods are the general family of methods that include: finite element (FE), spectral element (SE), and spectral methods. CG methods take complex geometries (elements) from physical space to computational space, using continuous basis functions that are used to approximate the solution to a given partial differential equation (PDE). To construct the problem relevant to this project, the generalized 2-D hyperbolic-elliptic PDE was first considered [7]:

$$\frac{\partial q}{\partial t} + \vec{u} \cdot \nabla q = \nu \nabla^2 q$$

where $q = q(\vec{x}, t)$, $\vec{u} = \vec{u}(\vec{x})$, $\vec{x} = (x, z)^T$, and ν is the viscosity coefficient. Using Galerkin machinery, q and $\vec{u}$ were approximated using basis function expansions:

$$q_N(\vec{x}, t) = \sum_{j=1}^{M_N} \psi_j(\vec{x}) q_j(t) \tag{2.25}$$

$$\vec{u}_N(\vec{x}, t) = \sum_{j=1}^{M_N} \psi_j(\vec{x}) \vec{u}_j(t) \tag{2.26}$$

where M_N represents the number of points inside the quadrilateral elements ($M_N = (N+1)^2$), N is the order of the polynomial approximation, and ψ_j is the Lagrange polynomial basis functions. The approximations for q_N and $\vec{u}_N$ were then substituted into the PDE, multiplied by a test function, ψ_I, and integrated across the global domain, Ω, to get (weak integral form):

$$\int_\Omega \psi_I \frac{\partial q_N}{\partial t} d\Omega + \int_\Omega \psi_I (\vec{u} \cdot \nabla q_N) d\Omega = \nu \int_\Omega \psi_I \nabla^2 q_N d\Omega \quad \forall \Psi \in H^1 \tag{2.27}$$

Instead of solving the global problem directly, 2-D local basis functions were constructed. The 2-D local basis functions are defined as:

$$\psi_i(\xi, \eta) = h_j(\xi) \otimes h_k(\eta)$$

$$j, k = 0, 1, ..., N \qquad i = 1, 2, ..., (N+1)^2$$

where h is a 1-D local basis function and $\otimes$ is the tensor/outer product of the 1-D local basis functions. The 1-D Lagrange polynomial local basis function, using Legendre-Gauss-Lobatto interpolation points, was defined by:

$$h_j(\xi) = \prod_{\substack{l=0 \\ l \neq j}}^{N} \left(\frac{\xi - \xi_l}{\xi_j - \xi_l} \right)$$

Additionally, in order to construct the basis functions and transition between physical space (x, z) and computational space (ξ, η) requires knowledge of the metric terms:

$$\frac{\partial \xi}{\partial x} = \frac{1}{|J|} \frac{\partial z}{\partial \eta} \quad , \quad \frac{\partial \xi}{\partial z} = \frac{-1}{|J|} \frac{\partial x}{\partial \eta}$$

$$\frac{\partial \eta}{\partial x} = \frac{-1}{|J|} \frac{\partial z}{\partial \xi} \quad , \quad \frac{\partial \eta}{\partial z} = \frac{1}{|J|} \frac{\partial x}{\partial \xi}$$

$$|J| = \frac{\partial x}{\partial \xi} \frac{\partial z}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial z}{\partial \xi}$$

With Eq. (2.27) in local form it can now be converted using integration by parts

15

(IBP) giving:

$$\int_{\Omega_e} \psi_i \nabla^2 q_N d\Omega_e = \int_{\Omega_e} \nabla \cdot (\psi_i \nabla q_N) d\Omega_e - \int_{\Omega_e} \nabla \psi_i \cdot \nabla (q_N) d\Omega_e \qquad (2.28)$$

then using the divergence theorem on Eq. (2.28) yields:

$$\int_{\Omega_e} \psi_i \nabla^2 q_N d\Omega_e = \int_{\Gamma_e} \vec{n} \cdot (\psi_i \nabla q_N) d\Gamma_e - \int_{\Omega_e} \nabla \psi_i \cdot \nabla (q_N) d\Omega_e \qquad (2.29)$$

The result from Eq. (2.29) is then substituted back into the original PDE seen in Eq. (2.27), which produces:

$$\int_{\Omega_e} \psi_i \frac{\partial q_N}{\partial t} d\Omega_e + \int_{\Omega_e} \psi_i (\vec{u} \cdot \nabla q_N) d\Omega_e = \nu \int_{\Gamma_e} \vec{n} \cdot (\psi_i \nabla q_N) d\Gamma_e - \nu \int_{\Omega_e} \nabla \psi_i \cdot \nabla (q_N) d\Omega_e$$

Substituting the summation approximation for q_N and $\vec{u}_N$ (Eqs. 2.25 and 2.26) yields:

$$\int_{\Omega_e} \psi_i \left(\sum_{j=1}^{M_N} \psi_j \frac{\partial q_j}{\partial t} \right) d\Omega_e + \int_{\Omega_e} \psi_i \left(\sum_{k=1}^{M_N} \psi_k \vec{u}_k \right) \cdot \left(\sum_{j=1}^{M_N} \nabla \psi_j q_j \right) d\Omega_e$$

$$= \nu \int_{\Gamma_e} \psi_i \vec{n} \cdot \left(\sum_{j=1}^{M_N} \nabla \psi_j q_j \right) d\Gamma_e - \nu \int_{\Omega_e} \nabla \psi_i \cdot \left(\sum_{j=1}^{M_N} \nabla \psi_j q_j \right) d\Omega_e$$

The resulting matrix problem is:

$$M_{ij}^{(e)} \frac{\partial q_j}{\partial t} + \vec{u}^T D_{ij}^{(e)} q_j = B_{ij}^{(e)} q_j - L_{ij}^{(e)} q_j$$

16

where $M_{ij}^{(e)}$ is the mass matrix, $D_{ij}^{(e)}$ is the differentiation matrix (a discrete representation of $\bar{\nabla}$), $B_{ij}^{(e)}$ is the boundary matrix, and $L_{ij}^{(e)}$ is the Laplacian matrix. The discretized matrices are:

$$M_{ij}^{(e)} = \sum_{l=1}^{M_N} \omega_l \, |J_l| \, \psi_{i,l} \psi_{j,l}$$

$$\vec{u}_k^T D_{ij}^{(e)} = \sum_{l=1}^{M_N} \omega_l \, |J_l| \, \psi_{i,l} \left(\sum_{k=1}^{M_N} \psi_{k,l} \vec{u}_k \right) \cdot \nabla \psi_{j,l}$$

$$B_{ij}^{(e)} = \sum_{l=1}^{Q+1} \omega_l \, |J_l| \, \psi_{i,l} \vec{n} \cdot \nabla \psi_{j,l}$$

$$L_{ij}^{(e)} = \sum_{l=1}^{M_N} \omega_l \, |J_l| \, \nabla \psi_{i,l} \cdot \nabla \psi_{j,l}$$

where Q represents the quadrature used for evaluation of the integrals, which when using inexact integration is equal to N. Direct stiffness summation (DSS) was then used to construct the global problem:

$$M_{IJ} = \bigwedge_{e=1}^{Ne} M_{ij}^{(e)}$$

$$L_{IJ} = \bigwedge_{e=1}^{Ne} L_{ij}^{(e)}$$

$$D_{IJ} = \bigwedge_{e=1}^{Ne} D_{ij}^{(e)}$$

where M_{IJ} reduces to a diagonal matrix M_I when using inexact integration.

1.   Equation Set 1

Applying these operators to the Navier-Stokes Equation Set 1 (2.2-2.3) yields:

$$\frac{\partial \pi_I}{\partial t} = -\vec{u}_I^T M_I^{-1} D_{IJ} \pi_J - \frac{R}{c_v} \pi_I M_I^{-1} D_{IJ}^T \vec{u}_J$$

$$\frac{\partial \vec{u}_I}{\partial t} = -\vec{u}_I^T M_I^{-1} D_{IJ} \vec{u}_J - c_p \theta M_I^{-1} D_{IJ} \pi_J - g M_I^{-1} \vec{k}$$

$$\frac{\partial \theta_I}{\partial t} = -\vec{u}_I^T M_I^{-1} D_{IJ} \theta_J$$

2.   Equation Set 2

Similarly to Set 1, the operators applied to the Navier-Stokes Equation Set 2 (2.4-2.6) yield:

$$\frac{\partial \rho_I}{\partial t} = -M_I^{-1} D_{IJ} (\rho \vec{u})_J$$

$$\frac{\partial \vec{u}_I}{\partial t} = -\vec{u}_I^T M_I^{-1} D_{IJ} \vec{u}_J - \frac{1}{\rho_I} M_I^{-1} D_{IJ} P_J - g M_I^{-1} \vec{k}$$

$$\frac{\partial \theta_I}{\partial t} = -\vec{u}_I^T M_I^{-1} D_{IJ} \theta_J$$

D. TEMPORAL DISCRETIZATION RUNGE-KUTTA 35

Integrating nonlinear PDEs forward in time can lead to numerous problems (i.e. spurious oscillations, overshoots, progressive smearing, etc...) if the proper time integration scheme is not used [8]. For the proposed test cases, a strong stability preserving (SSP) Runge-Kutta (RK) method was selected and implemented as outlined in Ruuth and Spiteri [8]. SSP time integration methods have strong nonlinear stability properties, which make them optimal, for this thesis, for temporal discretization because of the nonlinearities present in the Euler and Navier-Stokes equations. The particular method used for time integration was an explicit third-order five-stage RK method (RK35). This method was chosen for its large stability region relative to other explicit methods of equal order. The RK35 method can be represented by:

$$\boldsymbol{Q}_0 = \boldsymbol{Q}^{(n)}$$

$$\boldsymbol{Q}_I = \sum_{k=0}^{I-1} (\alpha_{I,k}\boldsymbol{Q}_k) + \beta_I \Delta t(RHS_{(I-1)}), \quad I = 1, 2, ..., s$$

$$\boldsymbol{Q}^{n+1} = \boldsymbol{Q}^{(s)}$$

where I is the stage (5 for RK35) and the coefficients $\alpha_{i,j}$ and β_i are listed in the Appendix.

19

THIS PAGE INTENTIONALLY LEFT BLANK

III.   APPLIED COORDINATE TRANSFORMS

A.   EQUATION SET 1

1.   Perturbation Method

This section will outline the expansion of the terms (application of the perturbation method) of set 1 of the Navier-Stokes Equations, where both π and θ will be split/decomposed into two components, the mean values ($\bar{\pi}$ and $\bar{\theta}$) and their associated perturbations (π' and θ') such that:

$$\pi = \bar{\pi}(z) + \pi'(x, z, t)$$

and

$$\theta = \bar{\theta}(z) + \theta'(x, z, t)$$

where the mean values satisfy a hydrostatically balanced atmosphere. After linearization, the pressure tendency Eq. (2.2) becomes:

$$\frac{\partial(\bar{\pi} + \pi')}{\partial t} + \vec{u} \cdot \nabla(\bar{\pi} + \pi') + \frac{R}{c_v}(\bar{\pi} + \pi')\nabla \cdot \vec{u} = 0$$

which, after simplification, becomes:

$$\frac{\partial \pi'}{\partial t} + \vec{u} \cdot \nabla \pi' + w\frac{\partial \bar{\pi}}{\partial z} + \frac{R}{c_v}(\bar{\pi} + \pi')\nabla \cdot \vec{u} = 0 \tag{3.1}$$

21

Expansion of the terms of the momentum Eq. (2.2) yields:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + c_p(\bar{\theta} + \theta')\nabla(\bar{\pi} + \pi') = -g\vec{k}$$

which can be expanded to:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + c_p(\bar{\theta} + \theta')\left[\left(\frac{\partial \bar{\pi}}{\partial x}, \frac{\partial \bar{\pi}}{\partial z}\right) + \left(\frac{\partial \pi'}{\partial x}, \frac{\partial \pi'}{\partial z}\right)\right] = -g\vec{k}$$

from which, using the hydrostatic approximation $\frac{d\bar{\pi}}{dz} = -\frac{g}{c_p\theta}$ yields:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + c_p(\bar{\theta} + \theta')\left[-\frac{g}{c_p\theta}\vec{k} + \left(\frac{\partial \pi'}{\partial x}, \frac{\partial \pi'}{\partial z}\right)\right] = -g\vec{k}$$

which can be simplified to:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + c_p(\bar{\theta} + \theta')\left(\frac{\partial \pi'}{\partial x}, \frac{\partial \pi'}{\partial z}\right) - g\vec{k} - g\frac{\theta'}{\theta}\vec{k} = -g\vec{k}$$

and finally becomes:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + c_p(\bar{\theta} + \theta')\nabla\pi' = g\frac{\theta'}{\theta}\vec{k} \tag{3.2}$$

Expansion of the terms of the thermodynamic energy Eq. (2.3) yields:

$$\frac{\partial(\bar{\theta} + \theta')}{\partial t} + \vec{u} \cdot \nabla(\bar{\theta} + \theta') = 0$$

22

which further leads to:

$$\frac{\partial \theta'}{\partial t} + \vec{u} \cdot \nabla \theta' + w\frac{\partial \bar{\theta}}{\partial z} = 0 \tag{3.3}$$

2.   Transform

Using the basic machinery prescribed in Eqs. (2.13) - (2.22), the set of non-conservative Navier-Stokes Eq. (3.1 - 3.3) were transformed from x-z coordinates to x-σ_z coordinates. The first machinery applied to the pressure tendency Eq. (3.1) was to change the vector dot products to transposes (i.e. $\vec{u} \cdot \nabla$ to $\vec{u}^T \nabla$), which yielded:

$$\frac{\partial \pi'}{\partial t} + \vec{u}^T \nabla \pi' + w\frac{\partial \bar{\pi}}{\partial z} + \frac{R}{c_v}(\bar{\pi} + \pi')\nabla \cdot \vec{u} = 0$$

substituting in Eq. (2.13) and Eq. (2.14) leads to:

$$\frac{\partial \pi'}{\partial t} + (J^{-1}\tilde{u})^T(J^T\tilde{\nabla})\pi' + w\frac{\partial \bar{\pi}}{\partial z}\frac{\partial \sigma_z}{\partial \sigma_z} + \frac{R}{c_v}(\bar{\pi} + \pi')(J^T\tilde{\nabla})^T(J^{-1}\tilde{u}) = 0$$

applying the linear algebra identity in Eq. (2.15) and the transformation for (w) yields:

$$\frac{\partial \pi'}{\partial t} + (\tilde{u})^T(J^{-1})^T(J^T)(\tilde{\nabla})\pi' + \left(-\tilde{u}\frac{\partial z_s}{\partial x}\frac{\sigma_z - H}{H} + \tilde{w}\frac{H - z_s}{H}\right)\frac{\partial \bar{\pi}}{\partial \sigma_z}\frac{\partial \sigma_z}{\partial z}$$

$$+\frac{R}{c_v}(\bar{\pi} + \pi')(\tilde{\nabla})^T(J^T)^T(J^{-1})(\tilde{u}) = 0$$

23

which simplifies to:

$$\frac{\partial \pi'}{\partial t} + (\tilde{\vec{u}})^T (\tilde{\nabla}) \pi' + \left(-\tilde{u} \frac{\partial z_s}{\partial x} \frac{\sigma_z - H}{H} + \tilde{w} \frac{H - z_s}{H} \right) \frac{\partial \bar{\pi}}{\partial \sigma_z} \left(\frac{H}{H - z_s} \right) + \frac{R}{c_v} (\bar{\pi} + \pi') (\tilde{\nabla})^T (\tilde{\vec{u}}) = 0$$

and then to:

$$\frac{\partial \pi'}{\partial t} + \tilde{\vec{u}} \cdot \tilde{\nabla} \pi' - \tilde{u} \left(\frac{\sigma_z - H}{H - z_s} \right) \frac{\partial z_s}{\partial x} \frac{\partial \bar{\pi}}{\partial \sigma_z} + \tilde{w} \frac{\partial \bar{\pi}}{\partial \sigma_z} + \frac{R}{c_v} (\bar{\pi} + \pi') \tilde{\nabla} \cdot \tilde{\vec{u}} = 0$$

and then further simplified to:

$$\frac{\partial \pi'}{\partial t} + \tilde{\vec{u}} \cdot \tilde{\nabla} \pi' - \tilde{u} \left(\frac{\sigma_z - H}{H - z_s} \right) \frac{\partial z_s}{\partial \sigma_z} \frac{\partial \bar{\pi}}{\partial x} + \tilde{w} \frac{\partial \bar{\pi}}{\partial \sigma_z} + \frac{R}{c_v} (\bar{\pi} + \pi') \tilde{\nabla} \cdot \tilde{\vec{u}} = 0$$

to finally yield:

$$\frac{\partial \pi'}{\partial t} + \tilde{\vec{u}} \cdot \tilde{\nabla} \pi' + \tilde{w} \frac{\partial \bar{\pi}}{\partial \sigma_z} + \frac{R}{c_v} (\bar{\pi} + \pi') \tilde{\nabla} \cdot \tilde{\vec{u}} = 0 \tag{3.4}$$

Applying the same machinery as above to the momentum Eq. (3.2), the dot products were replaced, which yields:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u}^T \nabla \vec{u} + c_p (\bar{\theta} + \theta') \nabla \pi' = g \frac{\theta'}{\bar{\theta}} \vec{k}$$

then $\vec{u}$ and ∇ were replaced by their transforms (Eq. (2.13) and Eq. (2.14)) leading to:

$$\frac{\partial \vec{u}}{\partial t} + (J^{-1} \tilde{\vec{u}})^T (J^T \tilde{\nabla}) \vec{u} + c_p (\bar{\theta} + \theta') (J^T \tilde{\nabla}) \pi' = g \frac{\theta'}{\bar{\theta}} \vec{k}$$

24

applying the linear algebra identity in Eq. (2.15) yields:

$$\frac{\partial \vec{u}}{\partial t} + (\tilde{\vec{u}})^T (J^{-1})^T (J^T)(\tilde{\nabla})\vec{u} + c_p(\bar{\theta} + \theta')(J^T \tilde{\nabla})\pi' = g\frac{\theta'}{\theta}\vec{k}$$

which simplifies to:

$$\frac{\partial \vec{u}}{\partial t} + (\tilde{\vec{u}})^T (\tilde{\nabla})\vec{u} + c_p(\bar{\theta} + \theta')(J^T \tilde{\nabla})\pi' = g\frac{\theta'}{\theta}\vec{k}$$

to finally yield:

$$\frac{\partial \vec{u}}{\partial t} + \tilde{\vec{u}} \cdot \tilde{\nabla}\vec{u} + c_p(\bar{\theta} + \theta')(J^T \tilde{\nabla})\pi' = g\frac{\theta'}{\theta}\vec{k} \qquad (3.5)$$

Applying the machinery to thermodynamic energy, Eq. (3.3) yields:

$$\frac{\partial \theta'}{\partial t} + \vec{u}^T \nabla\theta' + w\frac{\partial \bar{\theta}}{\partial z} = 0$$

substituting in Eq. (2.13) and Eq. (2.14) leads to:

$$\frac{\partial \theta'}{\partial t} + (J^{-1}\tilde{\vec{u}})^T (J^T \tilde{\nabla})\theta' + w\frac{\partial \bar{\theta}}{\partial z}\frac{\partial \sigma_z}{\partial \sigma_z} = 0$$

applying the linear algebra identity in Eq. (2.15) and the transformation for (w) yields:

$$\frac{\partial \theta'}{\partial t} + (\tilde{\vec{u}})^T (J^{-1})^T (J^T)(\tilde{\nabla})\theta' + \left(-\tilde{u}\frac{\partial z_s}{\partial x}\frac{\sigma_z - H}{H} + \tilde{w}\frac{H - z_s}{H}\right)\frac{\partial \bar{\theta}}{\partial \sigma_z}\frac{\partial \sigma_z}{\partial z} = 0$$

which simplifies to:

$$\frac{\partial \theta'}{\partial t} + (\tilde{\vec{u}})^T(\tilde{\nabla})\theta' + \left(-\tilde{u}\frac{\partial z_s}{\partial x}\frac{\sigma_z - H}{H} + \tilde{w}\frac{H - z_s}{H}\right)\frac{\partial \bar{\theta}}{\partial \sigma_z}\left(\frac{H}{H - z_s}\right) = 0$$

and simplifies further, yielding:

$$\frac{\partial \theta'}{\partial t} + \tilde{\vec{u}} \cdot \tilde{\nabla}\theta' - \tilde{u}\frac{\sigma_z - H}{H - z_s}\frac{\partial z_s}{\partial x}\frac{\partial \bar{\theta}}{\partial \sigma_z} + \tilde{w}\frac{\partial \bar{\theta}}{\partial \sigma_z} = 0$$

and then to:

$$\frac{\partial \theta'}{\partial t} + \tilde{\vec{u}} \cdot \tilde{\nabla}\theta' - \tilde{u}\frac{\sigma_z - H}{H - z_s}\frac{\partial z_s}{\partial \sigma_z}\frac{\partial \bar{\theta}}{\partial x} + \tilde{w}\frac{\partial \bar{\theta}}{\partial \sigma_z} = 0$$

to finally yield:

$$\frac{\partial \theta'}{\partial t} + \tilde{\vec{u}} \cdot \tilde{\nabla}\theta' + \tilde{w}\frac{\partial \bar{\theta}}{\partial \sigma_z} = 0 \tag{3.6}$$

3.   Decomposition

In order to discretize the governing equations and code them into Fortran 90/95, the vector fields had to be decomposed into scalar components. The decomposition of the pressure tendency Eq. (3.4) became:

$$\frac{\partial \pi'}{\partial t} + \left[\tilde{u}\frac{\partial \pi'}{\partial x} + \tilde{w}\frac{\partial \pi'}{\partial \sigma_z}\right] + \tilde{w}\frac{\partial \bar{\pi}}{\partial \sigma_z} + \frac{R}{c_v}(\bar{\pi} + \pi')\left[\frac{\partial \tilde{u}}{\partial x} + \frac{\partial \tilde{w}}{\partial \sigma_z}\right] = 0 \tag{3.7}$$

26

The momentum Eq. (3.5) decomposes into two separate equations (u and w), with the first equation (u) taking the form:

$$\frac{\partial u}{\partial t} + \left[\tilde{u}\frac{\partial u}{\partial x} + \tilde{w}\frac{\partial u}{\partial \sigma_z} \right] + c_p(\bar{\theta} + \theta') \left[\frac{\partial \pi'}{\partial x} + \left(\frac{\partial z_s}{\partial x}\frac{\sigma_z - H}{H - z_s} \right) \frac{\partial \pi'}{\partial \sigma_z} \right] = 0 \qquad (3.8)$$

and the representation for (w) taking the form:

$$\frac{\partial w}{\partial t} + \left[\tilde{u}\frac{\partial w}{\partial x} + \tilde{w}\frac{\partial w}{\partial \sigma_z} \right] + c_p(\bar{\theta} + \theta') \left[\left(\frac{H}{H - z_s} \right) \frac{\partial \pi'}{\partial \sigma_z} \right] = g\frac{\theta'}{\bar{\theta}} \qquad (3.9)$$

The decomposed thermodynamic energy Eq. (3.6) yields:

$$\frac{\partial \theta'}{\partial t} + \left[\tilde{u}\frac{\partial \theta'}{\partial x} + \tilde{w}\frac{\partial \theta'}{\partial \sigma_z} \right] + \tilde{w}\frac{\partial \bar{\theta}}{\partial \sigma_z} = 0 \qquad (3.10)$$

4.   Application of the Galerkin Statement

Using the Galerkin machinery outlined in the previous chapter, Eq. (3.7) - (3.10) were discretized yielding the continuous Galerkin Set 1 using x-σ_z coordinates (CG1 x-σ_z):

$$\sum_{l=1}^{M_N} \omega_l \left| J_l \right| \psi_{i,l}\psi_{j,l}\frac{\partial \pi'_j}{\partial t} =$$

$$\sum_{l=1}^{M_N} \omega_l \left| J_l \right| \psi_{i,l} \left[-\sum_{k=1}^{M_N} \psi_{k,l} \left(\tilde{u}_k\frac{\partial \psi_{j,l}}{\partial x}\pi'_j + \tilde{w}_k\frac{\partial \psi_{j,l}}{\partial \sigma_z}\pi'_j - \tilde{w}_k\frac{\partial \psi_{j,l}}{\partial \sigma_z}\bar{\pi}_j \right) \right.$$

$$\left. -\frac{R}{c_v} \left(\sum_{k=1}^{M_N} \psi_{k,l}(\bar{\pi}_k + \pi'_k) \right) \left(\frac{\partial \psi_{j,l}}{\partial x}\tilde{u}_j + \frac{\partial \psi_{j,l}}{\partial \sigma_z}\tilde{w}_j \right) \right] \qquad (3.11)$$

27

$$\sum_{l=1}^{M_N} \omega_l \left|J_l\right| \psi_{i,l}\psi_{j,l}\frac{\partial u_j}{\partial t} =$$

$$\sum_{l=1}^{M_N} \omega_l \left|J_l\right| \psi_{i,l} \left[-\sum_{k=1}^{M_N} \psi_{k,l}\left(\tilde{u}_k\frac{\partial \psi_{j,l}}{\partial x}u_j + \tilde{w}_k\frac{\partial \psi_{j,l}}{\partial \sigma_z}u_j \right) \right.$$

$$\left. -c_p\left(\sum_{k=1}^{M_N} \psi_{k,l}(\bar{\theta}_k + \theta'_k) \right)\left(\frac{\partial \psi_{j,l}}{\partial x}\pi'_j + \left[\left(\frac{\partial z_s}{\partial x} \right)_j \frac{\sigma_{z,j} - H}{H - z_{s,j}} \right]\frac{\partial \psi_{j,l}}{\partial \sigma_z}\pi'_j \right) \right] \qquad (3.12)$$

$$\sum_{l=1}^{M_N} \omega_l \left|J_l\right| \psi_{i,l}\psi_{j,l}\frac{\partial w_j}{\partial t} =$$

$$\sum_{l=1}^{M_N} \omega_l \left|J_l\right| \psi_{i,l} \left[-\sum_{k=1}^{M_N} \psi_{k,l}\left(\tilde{u}_k\frac{\partial \psi_{j,l}}{\partial x}w_j + \tilde{w}_k\frac{\partial \psi_{j,l}}{\partial \sigma_z}w_j \right) \right.$$

$$\left. -c_p\left(\sum_{k=1}^{M_N} \psi_{k,l}(\bar{\theta}_k + \theta'_k) \right)\left(\left[\frac{H}{H - z_{s,j}} \right]\frac{\partial \psi_{j,l}}{\partial \sigma_z}\pi'_j \right) + g\frac{\theta'_j}{\bar{\theta}_j} \right] \qquad (3.13)$$

$$\sum_{l=1}^{M_N} \omega_l \left|J_l\right| \psi_{i,l}\psi_{j,l}\frac{\partial \theta'_j}{\partial t} =$$

$$\sum_{l=1}^{M_N} \omega_l \left|J_l\right| \psi_{i,l} \left[-\sum_{k=1}^{M_N} \psi_{k,l}\left(\tilde{u}_k\frac{\partial \psi_{j,l}}{\partial x}\theta'_j + \tilde{w}_k\frac{\partial \psi_{j,l}}{\partial \sigma_z}\theta'_j \right) - \tilde{w}_j\frac{\partial \psi_{j,l}}{\partial \sigma_z}\bar{\theta}_j \right] \qquad (3.14)$$

The existing CG1 x-z code (in Fortran 90/95) [1] was modified using Eqs. (3.11) - (3.14) and then used for the test cases outlined in the next chapter.

B. EQUATION SET 2

1. Perturbation Method

This section will outline the expansion of the terms of set 2 of the Navier-Stokes Equations, where (as seen in set 1) both ρ and θ will be split/decomposed into two components, the reference values ($\bar{\rho}$ and $\bar{\theta}$) and their associated perturbations (ρ' and θ'). As seen in set 1:

$$\rho = \bar{\rho}(z) + \rho'(x, z, t)$$

and

$$\theta = \bar{\theta}(z) + \theta'(x, z, t)$$

where the reference values again satisfy a hydrostatically balanced atmosphere. In order to linearize, the mass Eq. (2.4) the product rule is first applied, yielding:

$$\frac{\partial \rho}{\partial t} + \vec{u} \cdot \nabla \rho + \rho \nabla \cdot \vec{u} = 0$$

followed by expansion of the terms, becomes:

$$\frac{\partial(\bar{\rho} + \rho')}{\partial t} + \vec{u} \cdot \nabla(\bar{\rho} + \rho') + (\bar{\rho} + \rho')\nabla \cdot \vec{u} = 0$$

which, after simplification, yields:

29

$$\frac{\partial \rho'}{\partial t} + \vec{u} \cdot \nabla \rho' + w \frac{\partial \bar{\rho}}{\partial z} + (\bar{\rho} + \rho')\nabla \cdot \vec{u} = 0 \qquad (3.15)$$

By setting $\mu = 0$, the momentum Eq. (2.6) becomes:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho}\nabla P = -g\vec{k}$$

and followed by an expansion of the terms, yields:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{(\bar{\rho} + \rho')}\nabla(\bar{P} + P') = -g\vec{k}$$

which leads to:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{(\bar{\rho} + \rho')}\nabla P' + \frac{1}{(\bar{\rho} + \rho')}\frac{\partial \bar{P}}{\partial z}\vec{k} = -g\vec{k}$$

from which, using the hydrostatic approximation $\frac{d\bar{\rho}}{dz} = -\bar{\rho}g$ yields:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{(\bar{\rho} + \rho')}\nabla P' - \frac{\bar{\rho}g}{(\bar{\rho} + \rho')}\vec{k} = -g\vec{k}$$

which, after simplification, becomes:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{(\bar{\rho} + \rho')}\nabla P' + \frac{\rho'g}{(\bar{\rho} + \rho')}\vec{k} = 0$$

and finally:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{(\bar{\rho} + \rho')} \nabla P' = -\frac{\rho' g}{(\bar{\rho} + \rho')} \vec{k} \tag{3.16}$$

By setting $\mu = 0$, the thermodynamic energy Eq. (2.6) becomes:

$$\frac{\partial \theta}{\partial t} + \vec{u} \cdot \nabla \theta = 0$$

and followed by an expansion of the terms, yields:

$$\frac{\partial (\bar{\theta} + \theta')}{\partial t} + \vec{u} \cdot \nabla (\bar{\theta} + \theta') = 0$$

which further leads to:

$$\frac{\partial \theta'}{\partial t} + \vec{u} \cdot \nabla \theta' + w \frac{\partial \bar{\theta}}{\partial z} = 0 \tag{3.17}$$

2.   Transform

Using the basic machinery described in Eq. (2.13) - (2.22), the set of non-conservative Navier-Stokes Eq. (3.15 - 3.17) were transformed from x-z coordinates to x-σ_z coordinates. The first machinery applied to the pressure tendency Eq. (3.15) was to change the vector dot products to transposes (i.e. $\vec{u} \cdot \nabla$ to $\vec{u}^T \nabla$), which yielded:

$$\frac{\partial \rho'}{\partial t} + \vec{u}^T \nabla \rho' + w \frac{\partial \bar{\rho}}{\partial z} + (\bar{\rho} + \rho') \nabla^T \vec{u} = 0$$

31

substituting in Eq. (2.13) and Eq. (2.14) leads to:

$$\frac{\partial \rho'}{\partial t} + (J^{-1}\tilde{u})^T (J^T \tilde{\nabla})\rho' + w\frac{\partial \bar{\rho}}{\partial z}\frac{\partial \sigma_z}{\partial \sigma_z} + (\bar{\rho}+\rho')(J^T \tilde{\nabla})^T J^{-1}\tilde{u} = 0$$

applying the linear algebra identity in Eq. (2.15) and the transformation for (w) yields:

$$\frac{\partial \rho'}{\partial t} + (\tilde{u})^T (J^{-1})^T (J^T \tilde{\nabla})\rho' + \left(-\tilde{u}\frac{\partial z_s}{\partial x}\frac{\sigma_z - H}{H} + \tilde{w}\frac{H - z_s}{H}\right)\frac{\partial \bar{\rho}}{\partial \sigma_z}\frac{\partial \sigma_z}{\partial z}$$

$$+(\bar{\rho}+\rho')(\tilde{\nabla})^T (J^T)^T J^{-1}\tilde{u} = 0$$

which simplifies to:

$$\frac{\partial \rho'}{\partial t} + (\tilde{u})^T \tilde{\nabla}\rho' + \left(-\tilde{u}\frac{\partial z_s}{\partial x}\frac{\sigma_z - H}{H} + \tilde{w}\frac{H - z_s}{H}\right)\frac{\partial \bar{\rho}}{\partial \sigma_z}\left(\frac{H}{H - z_s}\right) + (\bar{\rho}+\rho')(\tilde{\nabla})^T \tilde{u} = 0$$

to finally yield:

$$\frac{\partial \rho'}{\partial t} + \tilde{u}\cdot\tilde{\nabla}\rho' + \tilde{w}\frac{\partial \bar{\rho}}{\partial \sigma_z} + (\bar{\rho}+\rho')\tilde{\nabla}\cdot\tilde{u} = 0 \tag{3.18}$$

Applying the same machinery as above to the momentum Eq. (3.16), the dot products were replaced, which yields:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u}^T\nabla\vec{u} + \frac{1}{(\bar{\rho}+\rho')}\nabla P' = -\frac{\rho' g}{(\bar{\rho}+\rho')}\vec{k}$$

32

then $\vec{u}$ and ∇ were replaced by their transforms (Eq. (2.13) and Eq. (2.14)) leading to:

$$\frac{\partial \vec{u}}{\partial t} + (J^{-1}\tilde{u})^T J^T \tilde{\nabla}\vec{u} + \frac{1}{(\bar{\rho}+\rho')}J^T\tilde{\nabla}P' = -\frac{\rho' g}{(\bar{\rho}+\rho')}\vec{k}$$

applying the linear algebra identity in Eq. (2.15) yields:

$$\frac{\partial \vec{u}}{\partial t} + \tilde{u}^T(J^{-1})^T J^T \tilde{\nabla}\vec{u} + \frac{1}{(\bar{\rho}+\rho')}J^T\tilde{\nabla}P' = -\frac{\rho' g}{(\bar{\rho}+\rho')}\vec{k}$$

which simplifies to:

$$\frac{\partial \vec{u}}{\partial t} + \tilde{u}^T \tilde{\nabla}\vec{u} + \frac{1}{(\bar{\rho}+\rho')}J^T\tilde{\nabla}P' = -\frac{\rho' g}{(\bar{\rho}+\rho')}\vec{k}$$

to finally yield:

$$\frac{\partial \vec{u}}{\partial t} + \tilde{u} \cdot \tilde{\nabla}\vec{u} + \frac{1}{(\bar{\rho}+\rho')}J^T\tilde{\nabla}P' = -\frac{\rho' g}{(\bar{\rho}+\rho')}\vec{k} \qquad (3.19)$$

Applying the machinery to thermodynamic energy Eq. (3.17) yields:

$$\frac{\partial \theta'}{\partial t} + \vec{u}^T\nabla\theta' + w\frac{\partial \bar{\theta}}{\partial z} = 0$$

substituting in Eq. (2.13) and Eq. (2.14) leads to:

$$\frac{\partial \theta'}{\partial t} + (J^{-1}\tilde{u})^T J^T \tilde{\nabla}\theta' + w\frac{\partial \bar{\theta}}{\partial z}\frac{\partial \sigma_z}{\partial \sigma_z} = 0$$

applying the linear algebra identity in Eq. (2.15) and the transformation for (w) yields:

$$\frac{\partial \theta'}{\partial t} + (\tilde{\vec{u}})^T (J^{-1})^T J^T \tilde{\nabla} \theta' + \left(-\tilde{u} \frac{\partial z_s}{\partial x} \frac{\sigma_z - H}{H} + \tilde{w} \frac{H - z_s}{H} \right) \frac{\partial \bar{\theta}}{\partial \sigma_z} \frac{\partial \sigma_z}{\partial z} = 0$$

which simplifies to:

$$\frac{\partial \theta'}{\partial t} + (\tilde{\vec{u}})^T \tilde{\nabla} \theta' + \left(-\tilde{u} \frac{\partial z_s}{\partial x} \frac{\sigma_z - H}{H} + \tilde{w} \frac{H - z_s}{H} \right) \frac{\partial \bar{\theta}}{\partial \sigma_z} \left(\frac{H}{H - z_s} \right) = 0$$

to finally yield:

$$\frac{\partial \theta'}{\partial t} + \tilde{\vec{u}} \cdot \tilde{\nabla} \theta' + \tilde{w} \frac{\partial \bar{\theta}}{\partial \sigma_z} = 0 \qquad (3.20)$$

3.   Decomposition

Similar to CG1, in order to discretize the governing equations and code them into Fortran 90/95, they had to be decomposed into scalar components. The decomposition of the mass Eq. (3.18) became:

$$\frac{\partial \rho'}{\partial t} + \left[\tilde{u} \frac{\partial \rho'}{\partial \sigma_z} + \tilde{w} \frac{\partial \rho'}{\partial \sigma_z} \right] + \tilde{w} \frac{\partial \bar{\rho}}{\partial \sigma_z} + (\bar{\rho} + \rho') \left[\frac{\partial \tilde{u}}{\partial x} + \frac{\partial \tilde{w}}{\partial \sigma_z} \right] = 0 \qquad (3.21)$$

The momentum Eq. (3.19) decomposes into two separate equations (u and w), with

the first equation (u) taking the form:

$$\frac{\partial u}{\partial t} + \left[\tilde{u}\frac{\partial u}{\partial x} + \tilde{w}\frac{\partial u}{\partial \sigma_z}\right] + \frac{1}{(\bar{\rho}+\rho')}\left[\frac{\partial P'}{\partial x} + \left(\frac{\partial z_s}{\partial x}\frac{\sigma_z - H}{H - z_s}\right)\frac{\partial P'}{\partial \sigma_z}\right] = 0 \qquad (3.22)$$

and the representation for (w) taking the form:

$$\frac{\partial w}{\partial t} + \left[\tilde{u}\frac{\partial w}{\partial x} + \tilde{w}\frac{\partial w}{\partial \sigma_z}\right] + \frac{1}{(\bar{\rho}+\rho')}\left[\left(\frac{H}{H - z_s}\right)\frac{\partial P'}{\partial \sigma_z}\right] = -\frac{\rho' g}{(\bar{\rho}+\rho')} \qquad (3.23)$$

The decomposed thermodynamic energy Eq. (3.20) yields:

$$\frac{\partial \theta'}{\partial t} + \left[\tilde{u}\frac{\partial \theta'}{\partial x} + \tilde{w}\frac{\partial \theta'}{\partial \sigma_z}\right] + \tilde{w}\frac{\partial \bar{\theta}}{\partial \sigma_z} = 0 \qquad (3.24)$$

4. Application of the Galerkin Statement

Using the Galerkin machinery outlined in the previous chapter, Eqs. (3.21) - (3.24) were discretized yielding the continuous Galerkin Set 2 using x-σ_z coordinates (CG2 x-σ_z):

$$\sum_{l=1}^{M_N}\omega_l\,|J_l|\,\psi_{i,l}\psi_{j,l}\frac{\partial \rho'_j}{\partial t} =$$

$$\sum_{l=1}^{M_N}\omega_l\,|J_l|\,\psi_{i,l}\left[-\sum_{k=1}^{M_N}\psi_{k,l}\left(\tilde{u}_k\left[\frac{\partial \psi_{j,l}}{\partial x} + \left(\left(\frac{\partial z_s}{\partial x}\right)_j\frac{\sigma_{z,j} - H}{H - z_{s,j}}\right)\frac{\partial \psi_{j,l}}{\partial \sigma_z}\right]\rho'_j\right.\right.$$

$$\left.+\tilde{w}_k\left(\frac{H}{H - z_{s,j}}\right)\frac{\partial \psi_{j,l}}{\partial \sigma_z}\rho'_j - \tilde{w}_k\frac{\partial \psi_{j,l}}{\partial \sigma_z}\bar{\rho}_j\right)$$

$$\left.-\left(\sum_{k=1}^{M_N}\psi_{k,l}(\bar{\rho}_k + \rho'_k)\right)\left(\frac{\partial \psi_{j,l}}{\partial x}\tilde{u}_j + \frac{\partial \psi_{j,l}}{\partial \sigma_z}\tilde{w}_j\right)\right] \qquad (3.25)$$

$$\sum_{l=1}^{M_N} \omega_l \left| J_l \right| \psi_{i,l} \psi_{j,l} \frac{\partial u_j}{\partial t} =$$

$$\sum_{l=1}^{M_N} \omega_l \left| J_l \right| \psi_{i,l} \left[-\sum_{k=1}^{M_N} \psi_{k,l} \left(\tilde{u}_k \frac{\partial \psi_{j,l}}{\partial x} u_j + \tilde{w}_k \frac{\partial \psi_{j,l}}{\partial \sigma_z} u_j \right) \right.$$

$$-\left(\sum_{k=1}^{M_N} \psi_{k,l} \left(\frac{1}{(\bar{\rho}_j + \rho_j')} \right) \right) \left(\frac{\partial \psi_{j,l}}{\partial x} P_j' + \left[\left(\frac{\partial z_s}{\partial x} \right)_j \frac{\sigma_{z,j} - H}{H - z_{s,j}} \right] \frac{\partial \psi_{j,l}}{\partial \sigma_z} P_j' \right) \right] \qquad (3.26)$$

$$\sum_{l=1}^{M_N} \omega_l \left| J_l \right| \psi_{i,l} \psi_{j,l} \frac{\partial w_j}{\partial t} =$$

$$\sum_{l=1}^{M_N} \omega_l \left| J_l \right| \psi_{i,l} \left[-\sum_{k=1}^{M_N} \psi_{k,l} \left(\tilde{u}_k \frac{\partial \psi_{j,l}}{\partial x} w_j + \tilde{w}_k \frac{\partial \psi_{j,l}}{\partial \sigma_z} w_j \right) \right.$$

$$-\left(\sum_{k=1}^{M_N} \psi_{k,l} \left(\frac{1}{(\bar{\rho}_j + \rho_j')} \right) \right) \left(\left[\frac{H}{H - z_{s,j}} \right] \frac{\partial \psi_{j,l}}{\partial \sigma_z} P_j' \right) - \frac{\rho_j' g}{(\bar{\rho}_j + \rho_j')} \right] \qquad (3.27)$$

$$\sum_{l=1}^{M_N} \omega_l \left| J_l \right| \psi_{i,l} \psi_{j,l} \frac{\partial \theta_j'}{\partial t} =$$

$$\sum_{l=1}^{M_N} \omega_l \left| J_l \right| \psi_{i,l} \left[-\sum_{k=1}^{M_N} \psi_{k,l} \left(\tilde{u}_k \frac{\partial \psi_{j,l}}{\partial x} \theta_j' + \tilde{w}_k \frac{\partial \psi_{j,l}}{\partial \sigma_z} \theta_j' \right) - \tilde{w}_j \frac{\partial \psi_{j,l}}{\partial \sigma_z} \bar{\theta}_j \right] \qquad (3.28)$$

The existing CG2 x-z code (in Fortran 90/95) was modified using Eq. (3.25) - (3.28) and then used for the test cases outlined in the next chapter.

IV.   TEST CASES

A.   OVERVIEW

Three specific cases where chosen to evaluate the σ-coordinate transformations: the rising thermal bubble, the linear hydrostatic mountain, and the linear non-hydrostatic mountain. The rising thermal bubble, case 1, was chosen as a benchmark in order to insure that in the absence of terrain the model dynamics still functioned properly (i.e. the σ-coordinate transformed equations reduce to the original governing equations) yielding the same results as the unmodified source code [1]. The linear hydrostatic mountain, case 2, and the linear non-hydrostatic mountain, case 3, were chosen to evaluate the advantages and disadvantages of the x-σ governing equation in relation to their x-z formulation in both hydrostatic and non-hydrostatic environments using the same terrain. This chapter will outline the test case assumptions, additional information derived for the σ-coordinates, and previous results that will be used for comparison.

B.   CASE 1: RISING THERMAL BUBBLE

This test case represents a highly nonlinear non-hydrostatic flow problem, by demonstrating the evolution of an initially at rest warm air bubble, relative to the surrounding environment, in a hydrostatic constant potential temperature environment as resolved by the Navier-Stokes equations. As the warm air bubble rises, it will deform, taking the shape of a mushroom cloud, due to the shearing motion caused by the resulting differential vertical velocity field. The initial distribution of potential temperature perturbations in order to drive vertical motion is defined by:

$$
\theta' = \begin{cases} 0 & for \ \ r > r_c, \\ \frac{\theta_c}{2}\left[1 + cos\left(\frac{\pi_c r}{r_c}\right)\right] & for \ \ r \le r_c, \end{cases}
$$

where $\theta_c = 0.5^oC$, π_c is the trigonometric constant, $r = \sqrt{(x - x_c)^2 + (z - z_c)^2}$ with the constants: $\bar{\theta} = 300$ K, $r_c = 250$ m, and $(x_c, z_c) = (500,350)$ m. [1] The domain of interest is $(x,z) \in [0, 1000]^2$ m integrated over the time interval, $t \in [0, 700]$ s. Additionally, the boundary conditions for all boundaries are no-flux.

The utility of running this case is to insure that the σ transformed governing equations yield the same results as the x-z set of the governing equations when the terrain is flat:

$$z_{surf} = 0$$

where z_{surf} is the height of the surface. The surface heights are vital in deriving the σ-coordinates (see equation 2.16). The slope of the surface must also be calculated for the transformation, which in this case leads to:

$$\frac{\partial z_{surf}}{\partial x} = 0$$

With a flat surface in combination with a slope of 0, the transformation of the coordinate system $(x\text{-}z_\sigma)$ reduces to x-z, making both the modified and unmodified codes yield identical results.

The unmodified source code was run in order to establish a baseline to evaluate the transformed source code. Three resolutions were used: 20 m (2601 grid points), 10 m (10201 grid points), and 5 m (40401 grid points). The resulting data for potential temperature perturbations was then plotted for each resolution after 700 s of model integration (see Figures 2 and 3). As seen in Figures 2 and 3, the warm air bubble did deform into a mushroom type formation, while maintaining symmetry.
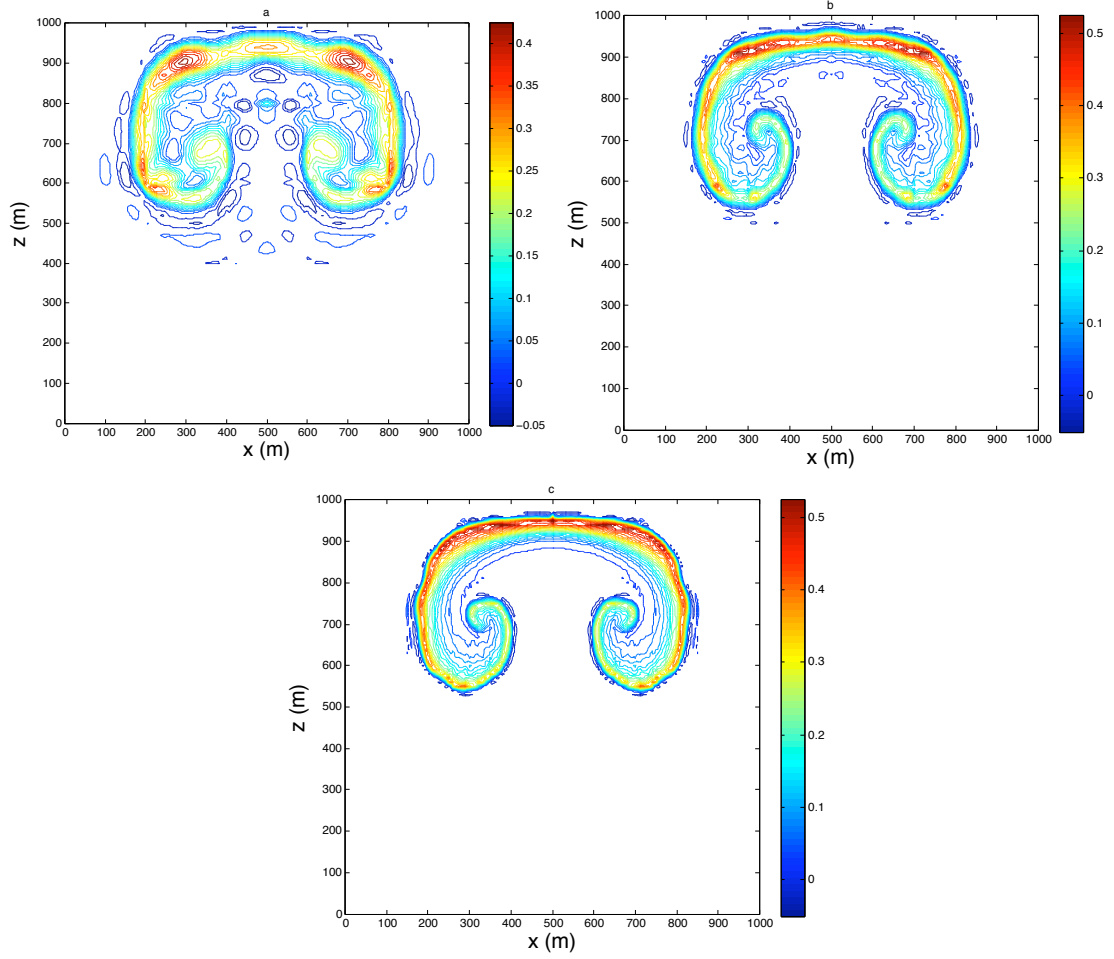
Figure 2. Case 1: Rising Thermal Bubble. Resulting potential temperature perturbations using unmodified CG1 source codes [1] after 700 s for resolutions: (a) 20, (b) 10, and (c) 5 m. All cases were run using 10th order polynomials, with contours from -0.05 to 0.525 K with an interval of 0.025 K.
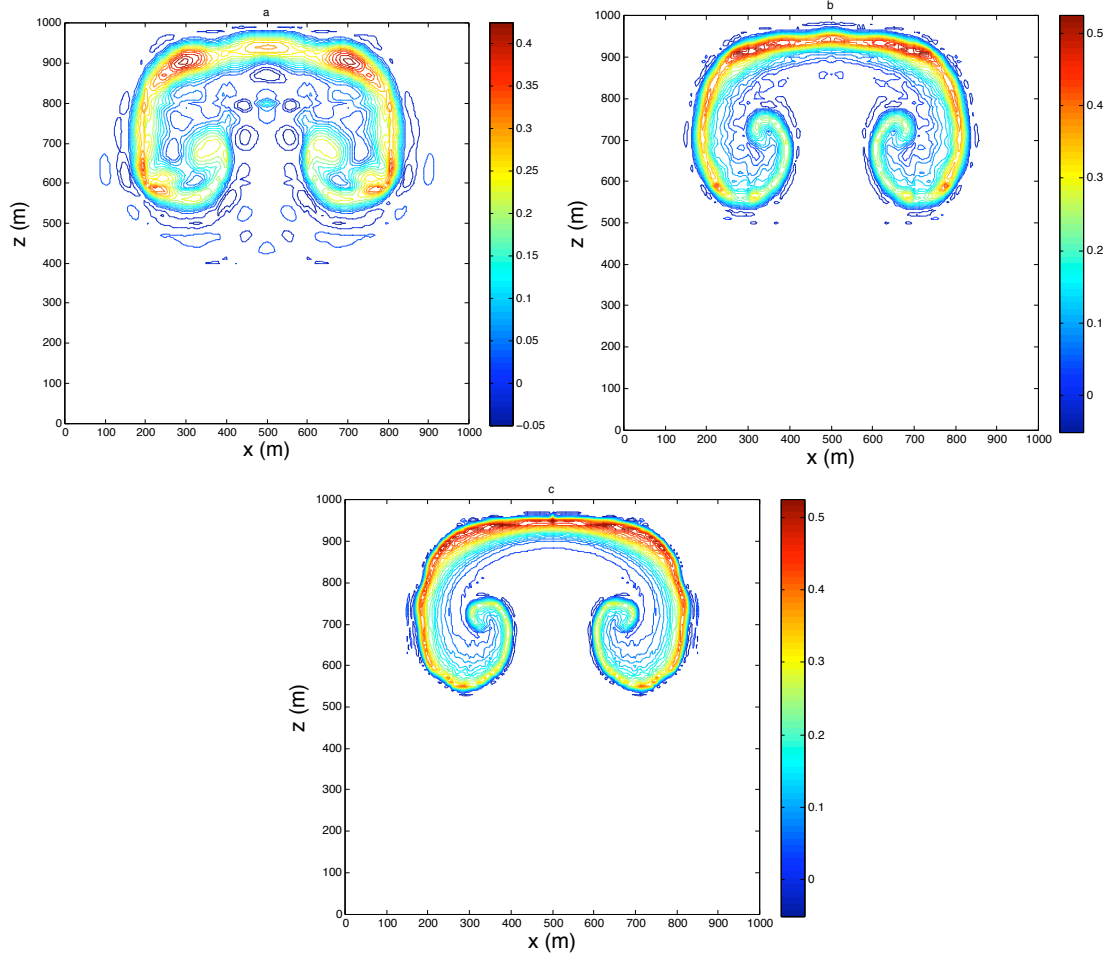
Figure 3. Case 1: Rising Thermal Bubble. Resulting potential temperature perturbations using unmodified CG2 source codes [1] after 700 s for resolutions: (a) 20, (b) 10, and (c) 5 m. All cases were run using 10th order polynomials, with contours from -0.05 to 0.525 K with an interval of 0.025 K.

C. CASE 2: LINEAR HYDROSTATIC MOUNTAIN

This test case was chosen in order to evaluate the model performance when a simple terrain feature is introduced in a linear hydrostatic flow environment with steady inflow and outflow lateral boundary conditions. Over time, a steady-state mountain wave over a single peak should form if the model dynamics are resolving the scenario accurately. Initially, the constant mean horizontal velocity is set to $\bar{u} = 20$ m/s, the mean vertical velocity is set to $\bar{w} = 0$ m/s, and the atmosphere is set to isothermal with a constant mean temperature of $\bar{T} = 250$ K.[1] Since this case is isothermal, the buoyancy frequency or Brunt-Väisälä frequency $\mathcal{N}^2 = g\frac{d}{dz}(ln\bar{\theta})$ reduces to $\mathcal{N} = \frac{g}{\sqrt{c_p\bar{T}}}$, which simplifies the Exner pressure to:

$$\bar{\pi} = e^{-\frac{g}{c_p\bar{T}}}$$

The domain of interest is (x,z) $\in$ [0, 240,000] $\times$ [0, 30,000] m integrated over the time interval, t $\in$ [0, 10] h. For this test case, the terrain (the *versiera di Agnesi* mountain profile) is represented by:

$$z_{surf} = \frac{hc}{\left(1 + \left(\frac{x-xc}{ac}\right)^2\right)}$$

where z_{surf} is the height of the surface, and h_c, x_c, and a_c are constants ($h_c = 1$ m, x_c = 120,000 m, and a_c = 10,000 m). Additionally, the bottom boundary conditions are no-flux, while the top and lateral boundaries use a non-reflecting boundary condition [1]. For further information on such boundary conditions, see the papers by Durran and Klemp [9], Giraldo and Restelli [1], and Dea, Giraldo, and Neta [10]. However non-reflecting boundary conditions are outside the scope of this thesis and will not be mentioned further. Unlike the previous case, the surface heights do vary with x and are even more vital in deriving the σ-coordinates. The surface plot can be seen in

Figure 4.a, on which it can be seen (noticing the z-axis scale) that the peak is fairly low and has smooth geometry. Since the surface height does vary with x, the slope of the surface must also be calculated for the transformation, which in this case leads to:

$$z_{surf} = hc \left(1 + \left(\frac{x - xc}{ac}\right)^2\right)^{-1}$$

$$z_{surf} = hc \left[\frac{(ac)^2 + (x^2 - 2(xc)(x) + (xc)^2)}{(ac)^2}\right]^{-1}$$

$$z_{surf} = (hc)(ac)^2 \left[(ac)^2 + x^2 - 2(xc)(x) + (xc)^2\right]^{-1}$$

$$\frac{\partial z_{surf}}{\partial x} = \frac{(-1)(hc)(ac)^2(2x - 2(xc))}{[(ac)^2 + x^2 - 2(xc)(x) + (xc)^2]^2}$$

$$\frac{\partial z_{surf}}{\partial x} = \frac{(-2)(hc)(ac)^2(x - xc)}{[(ac)^2 + (x - xc)^2]^2}$$

where the surface slope $\left(\frac{\partial z_{surf}}{\partial x}\right)$ plot can be seen in Figure 4.b, on which it can be seen (noticing the z-axis scale) that the slope of the peak is fairly flat and again has smooth geometry.

The unmodified source code was run (colored lines) and compared with the analytic solutions (black lines), generated for the unmodified source code and developed by Giraldo and Restelli [1], in order to establish a baseline to evaluate the transformed source code. One resolution was used: 1200 m (in x) and 240 m (in z) (20301 grid points). The resulting data for vertical and horizontal velocity was then plotted for the resolution after 10 h (see Figure 5). As seen in Figure 5, a steady-state mountain wave did form in the appropriate region and will be compared against the transformed governing equation sets.
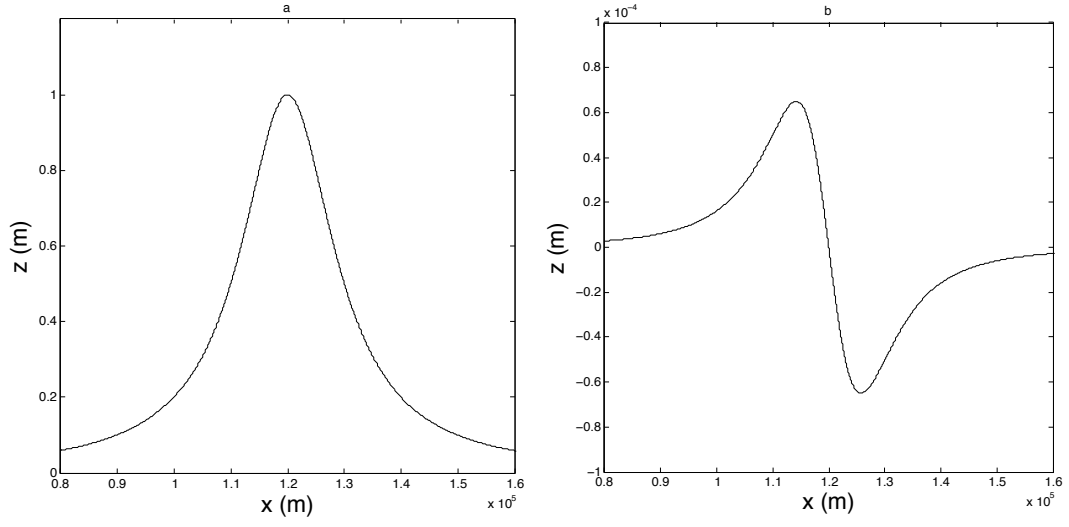
Figure 4. Case 2: Linear Hydrostatic Mountain. The single mountain peak z_{surf}, (a), and the associated slope of the peak $\frac{\partial z_{surf}}{\partial x}$, (b).
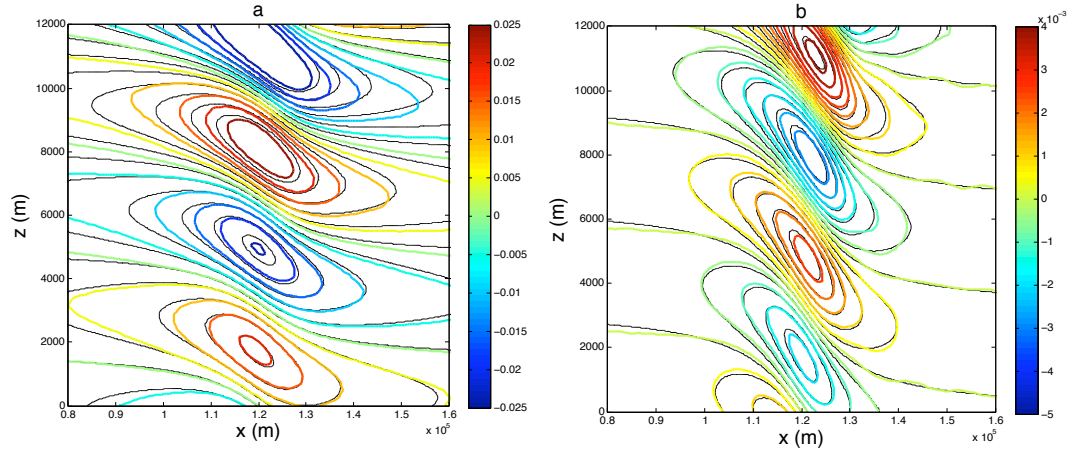
i)
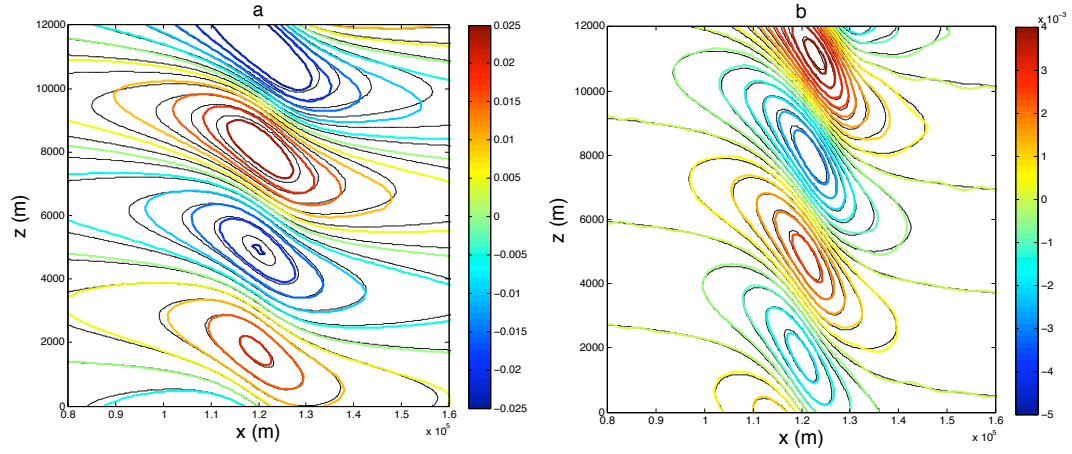
ii)

Figure 5. Case 2: Linear Hydrostatic Mountain. Resulting horizontal velocity (a) and vertical velocity (b) using unmodified (i) CG1 and (ii) CG2 source codes [1] after 10 h for the resolution of 1200 m (in x) and 240 m (in z) (colored lines) plotted with the analytic solution (black lines). All cases were run using 10th order polynomials, with contours from -0.025 to 0.025 ms^{-1} with an interval of 0.005 ms^{-1}, (a), and -0.005 to 0.005 ms^{-1} with an interval of 0.0005 ms^{-1}, (b).

D.   CASE 3: LINEAR NON-HYDROSTATIC MOUNTAIN

The final test case was chosen in order to evaluate the model performance when a simple terrain feature is introduced in a linear non-hydrostatic flow environment with steady inflow and outflow lateral boundary conditions. Over time, a steady-state mountain wave over a single peak should form if the model dynamics are resolving the scenario accurately. Initially, the constant mean horizontal velocity is set to $\bar{u} = 10$ m/s, the mean vertical velocity is set to $\bar{w} = 0$ m/s, and the atmosphere is uniformly stratified with a Brunt-Väisälä frequency of $\mathcal{N} = 0.01$ /s. This test case uses the same terrain as case 2 (see Figure 4). The constants used for this case were $h_c = 1$ m, $x_c = 72{,}000$ m, $a_c = 1{,}000$ m, and $\theta_0 = 280$K. The domain of interest is $(x,z) \in [0, 144{,}000] \times [0, 30{,}000]$ m integrated over the time interval, $t \in [0,5]$ h. Like case 2, the bottom boundary conditions are no-flux and the top and lateral boundaries are non-reflecting.[1]

The unmodified source code was again run (colored lines) and compared with the analytic solution [1] (black lines) in order to establish a baseline to evaluate the transformed source code. One resolution was used: 360 m (in x) and 300 m (in z) (40501 grid points). The resulting data for vertical and horizontal velocity was then plotted after 5 h (see Figure 6). As seen in Figure 6, a steady-state mountain wave did form in the appropriate region and will be compared against the transformed governing equation sets.
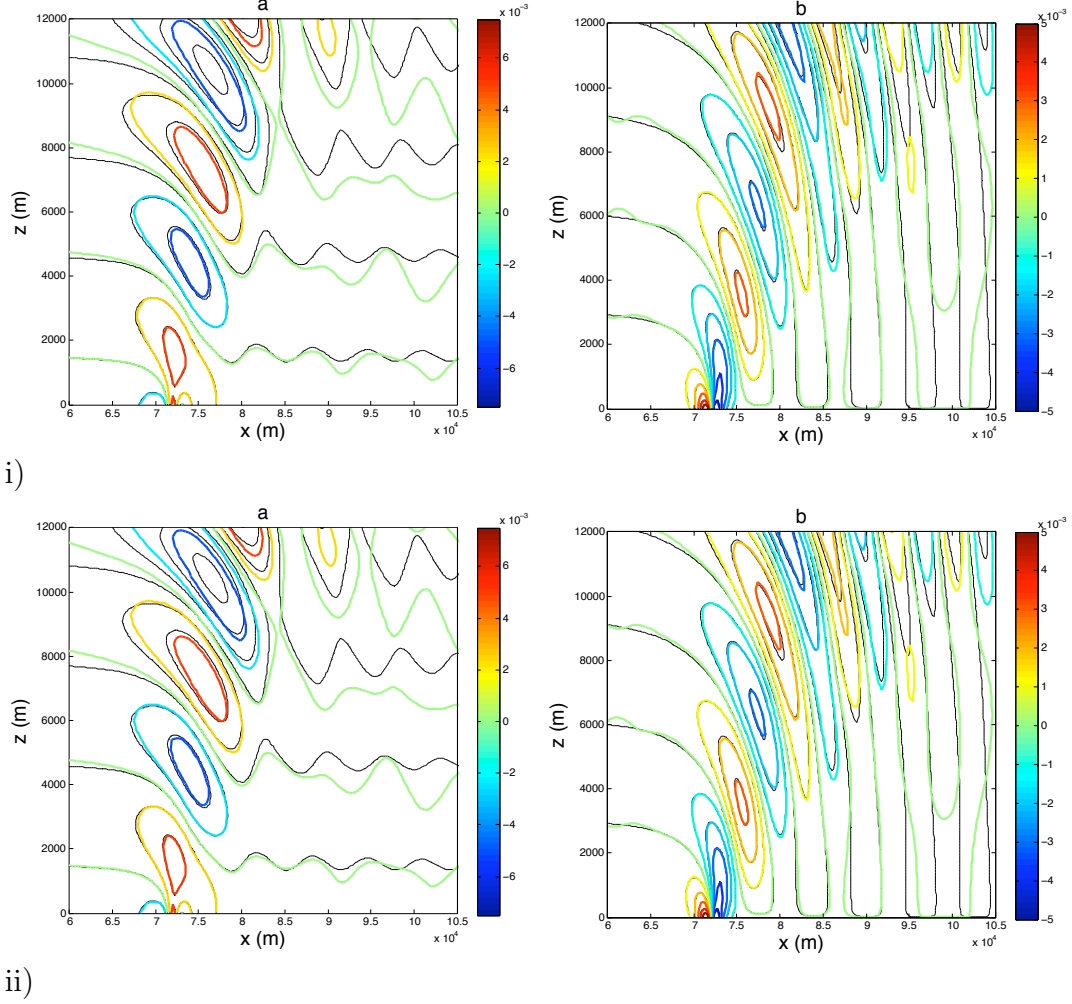
Figure 6. Case 3: Linear Non-Hydrostatic Mountain. Resulting horizontal velocity (a) and vertical velocity (b) using unmodified (i) CG1 and (ii) CG2 source codes [1] after 5 h for the resolution of 360 m (in x) and 300 m (in z) (colored lines) plotted with the analytic solution (black lines). All cases were run using 10th order polynomials, with contours from -0.025 to 0.025 ms^{-1} with an interval of 0.005 ms^{-1}, (a), and -0.005 to 0.005 ms^{-1} with an interval of 0.0005 ms^{-1}, (b).

V.   RESULTS

A.   CASE 1: RISING THERMAL BUBBLE

1.   Accuracy and Comparison

The modified source code (containing the σ-coordinate transformed governing equations) was run using the same three resolutions that were used for the unmodified code: 20 m (2601 grid points), 10 m (10201 grid points), and 5 m (40401 grid points). The resulting data for potential temperature was then plotted for each resolution after integrating forward 700 s (see Figure 7). As seen in Figure 7, the warm air bubble did deform into a mushroom type formation, while maintaining symmetry, appearing to duplicate the results seen by the x-z code (see figure 2).

To further evaluate the similarities in the x-z and x-σ_z model solutions nine variables (using 5 m resolution) were compared: the maximum Exner pressure perturbation (π'_{max} (unitless)), the minimum Exner pressure perturbation (π'_{min} (unitless)), the maximum horizontal wind velocity (u_{max} (ms^{-1})), the minimum horizontal wind velocity (u_{min} (ms^{-1})), the maximum vertical wind velocity (w_{max} (ms^{-1})), the minimum vertical wind velocity (w_{min} (ms^{-1})), the maximum potential temperature perturbation (θ'_{max} (K)), the minimum potential temperature perturbation (θ'_{min} (K)), and CPU time (s). These values were compare to better observe the relative maximum and minimum values occurring for each model run. In order to have consistent CPU constraints both x-z and x-σ_z for set 1 code were run in parallel on the same machine (NPS Math Department 32 Processor Apple Cluster: Riemann) and started at the same time, and then repeated for set 2. The resulting values can be seen in Table I. All four sets of code converge to nearly identical solutions. The only difference are the following: minor differences in the π'_{max} for set 1 and set 2; and differences in θ'_{max} and θ'_{min} between x-z coordinates and x-σ_z coordinates for both sets. The major difference between the models is the CPU time (in seconds), which was consistently

slower for the x-σ_z (28.24 % increase in time for set 1 and 15.58 % increase in time for set 2) relative to their x-z counterparts. This difference is caused by the additional x-σ_z data being passed between program functions and additional calculations being performed ($O(5 * [N_T] * [N_E] * [M_N]^2 * [8 * (N + 1) + 26])$), where N_T is the number of time steps, N_E is the number of elements, and M_N is the number of quadrature points or order $N+1$). For the 5 m resolution case, there were 5.2101×10^{19} additional operations. The horizontal velocities u_{max} and u_{min} also indicate that symmetry is maintained for all four model sets.

To further evaluate the performance of the four model runs, the Root Mean Squared Error (RMSE) was determined for each run in relation to the analytic solution (see Table II) for the four variables: π, u, w, and θ. The RMSE was calculated using:

$$\|\vec{q}\,\|_{RMS} = \sqrt{\frac{1}{N_p} \sum_{i=1}^{N_p} (\vec{q}^{\;numerical} - \vec{q}^{\;analytic})^2}$$

where $\vec{q}$ represents the given state variable vector, N_p is the number of points, and the analytic solution for this case was defined by the unmodified source code numerical

Table I. Case 1: Rising Thermal Bubble. Comparison of modified and unmodified CG1 and CG2 using 5 m resolution after 700s.

Model	CG1 x-z	CG1 x-σ_z	CG2 x-z	CG2 x-σ_z
π'_{max}	0.9364×10^{-5}	0.9364×10^{-5}	0.9365×10^{-5}	0.9365×10^{-5}
π'_{min}	-0.1195×10^{-4}	-0.1195×10^{-4}	-0.1195×10^{-4}	-0.1195×10^{-4}
u_{max} (ms^{-1})	0.2079×10^{1}	0.2079×10^{1}	0.2079×10^{1}	0.2079×10^{1}
u_{min} (ms^{-1})	-0.2079×10^{1}	-0.2079×10^{1}	-0.2079×10^{1}	-0.2079×10^{1}
w_{max} (ms^{-1})	0.2536×10^{1}	0.2536×10^{1}	0.2536×10^{1}	0.2536×10^{1}
w_{min} (ms^{-1})	-0.1912×10^{1}	-0.1912×10^{1}	-0.1912×10^{1}	-0.1912×10^{1}
θ'_{max} (K)	0.5713×10^{0}	0.5715×10^{0}	0.5713×10^{0}	0.5715×10^{0}
θ'_{min} (K)	-0.9736×10^{-1}	-0.9735×10^{-1}	-0.9736×10^{-1}	-0.9735×10^{-1}
CPU time (s)	0.1027×10^{5}	0.1317×10^{5}	0.1322×10^{5}	0.1528×10^{5}

Table II. Case 1: Rising Thermal Bubble RMSE. Root-mean-squared errors for the four primary state variables, for the modified codes using the unmodified code solutions as the analytic solutions, after 700 s using 5 m resolution and 10th order polynomials.

Model	π	u (ms^{-1})	w (ms^{-1})	θ (K)
CG1 x-σ_z	2.5931×10^{-15}	9.2434×10^{-10}	1.6068×10^{-9}	1.8639×10^{-9}
CG2 x-σ_z	2.2034×10^{-16}	5.2908×10^{-12}	8.5529×10^{-12}	9.6233×10^{-12}

solutions in order to better compare x-z solution to the x-σ_z solution. Both sets 1 and 2 exhibited a smaller error for all four variables of interest, with set 2 having the lower RMSE values overall.

In addition to comparing extremes of the selected fields, a one-dimensional vertical temperature profile along the centerline (x = 500 m) was plotted for each resolution to better discern the differences between the various models (see Figure 9). As the resolution increased from 20 m to 5 m, there is a noticeable increase in the sharpness of the temperature distributions. As expected, for all three resolutions, there is no significant difference between the temperatures along the centerline for each model at that specific resolution.

2. Conclusions

All four model runs for the rising thermal bubble did deform into a mushroom type formation, while maintaining symmetry (as indicated by the distribution of horizontal velocities). Additionally, the models converged to nearly identical solutions, as demonstrated by the *min* and *max* values and the RMSE. Overall, for both sets 1 and 2 of the Navier-Stokes equations, the x-σ_z coordinates do reduce to x-z, showing that in the absence of terrain that the model dynamic still function properly, but did show a noticeable increase in computational expense. These results validate the first stage of the code evaluation.
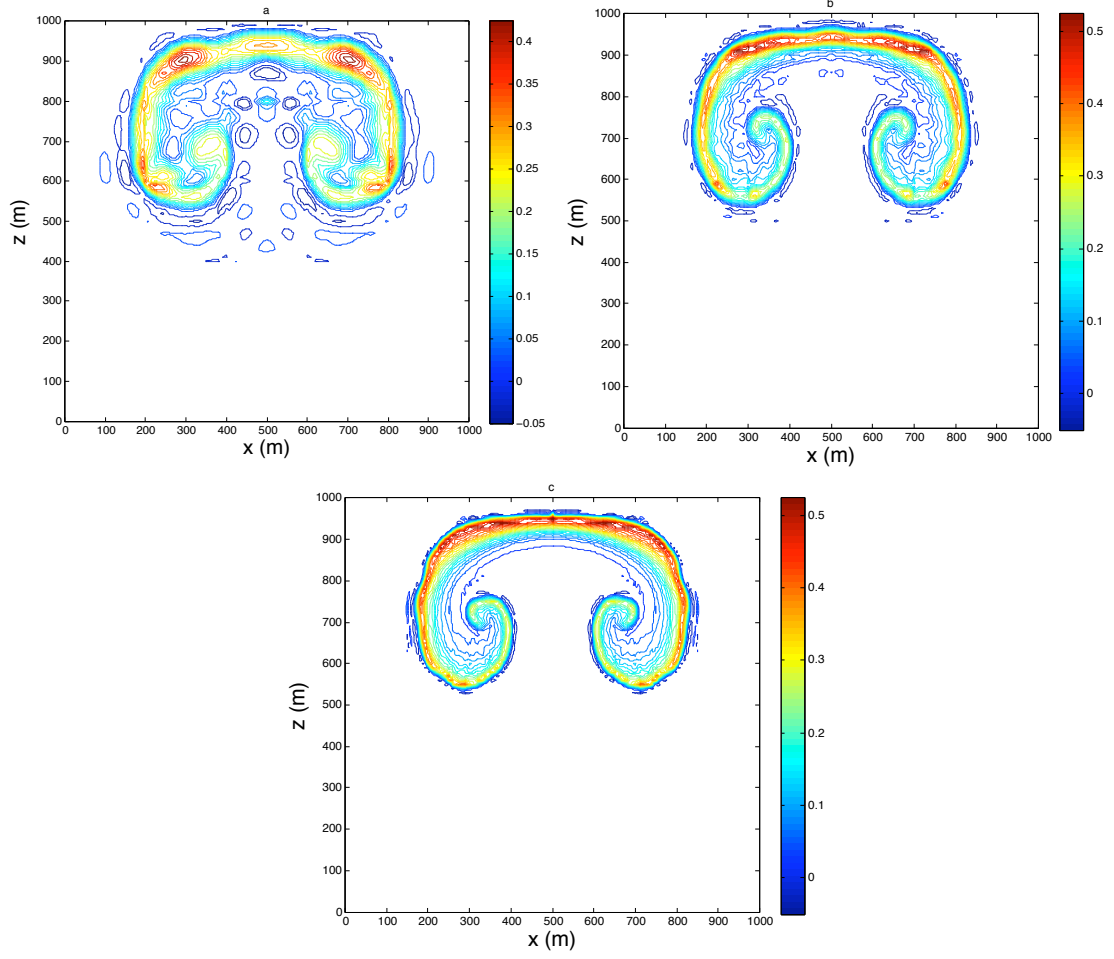
Figure 7. Case 1: Rising Thermal Bubble. Resulting potential temperature perturbations using modified CG1 source codes [1] after 700 s for resolutions: (a) 20, (b) 10, and (c) 5 m. All cases were run using 10th order polynomials, with contours from -0.05 to 0.525 K with an interval of 0.025 K.

Figure 8. Case 1: Rising Thermal Bubble. Resulting potential temperature perturbations using modified CG2 source codes [1] after 700 s for resolutions: (a) 20, (b) 10, and (c) 5 m. All cases were run using 10th order polynomials, with contours from -0.05 to 0.525 K with an interval of 0.025 K.

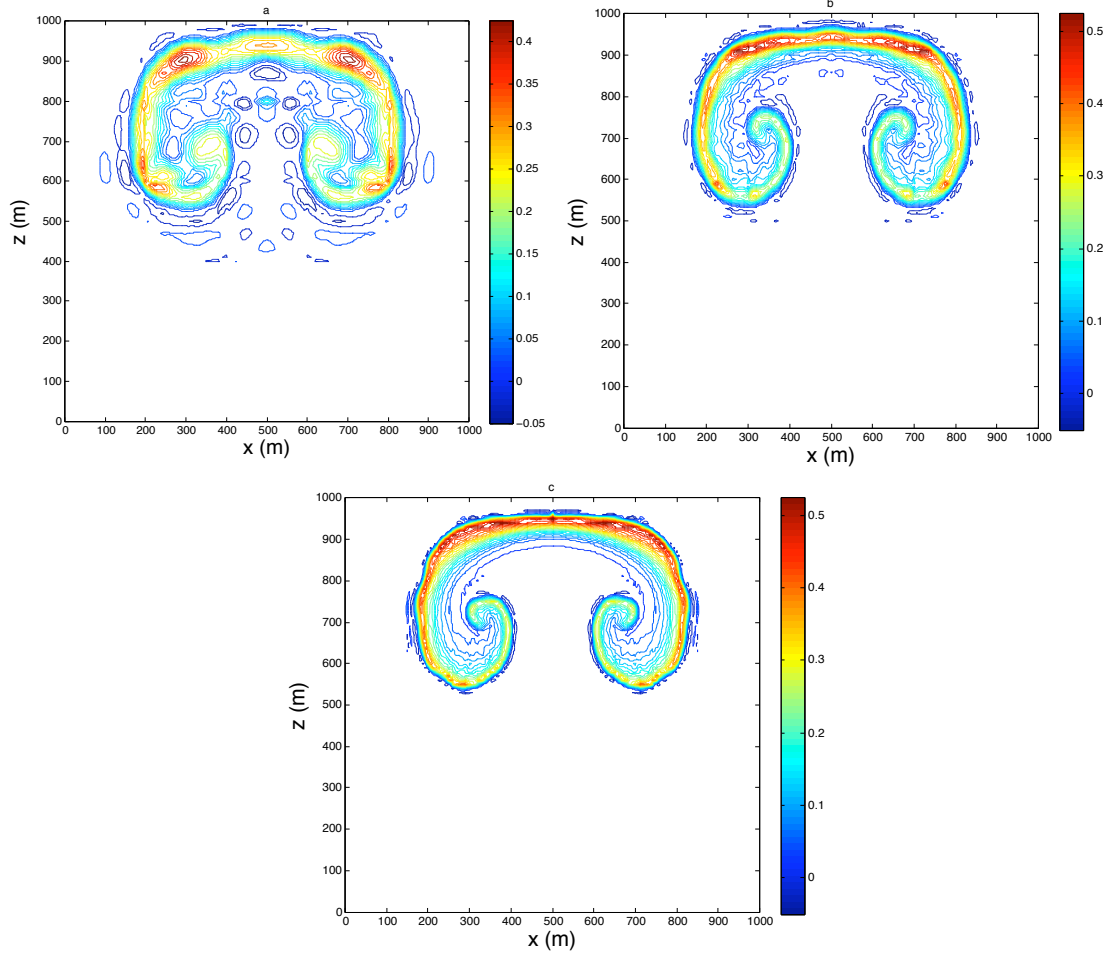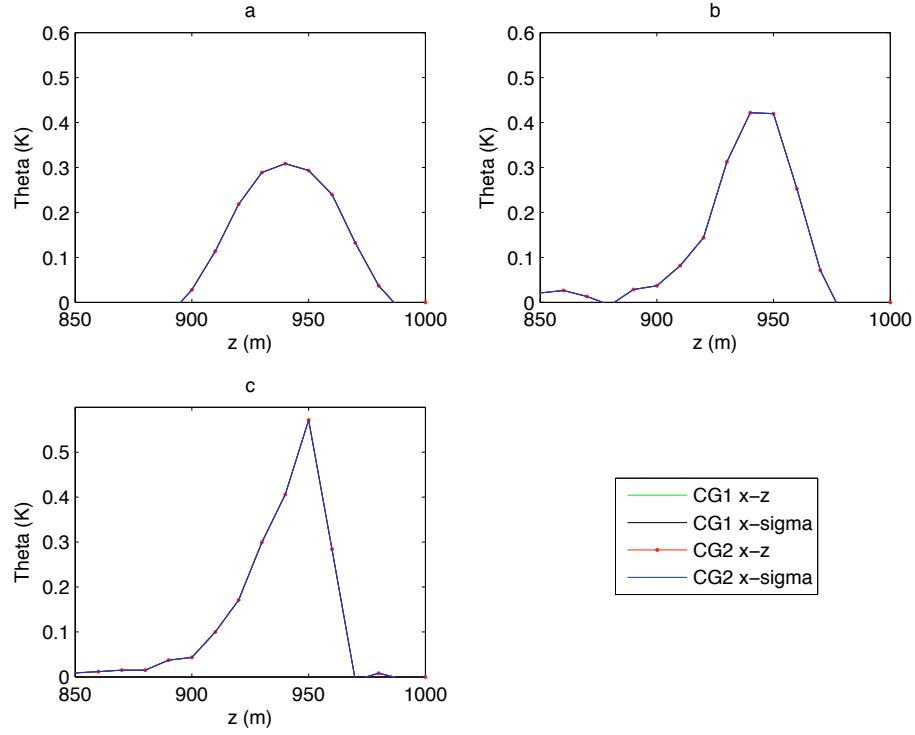Figure 9. Case 1: Rising Thermal Bubble. Resulting potential temperature perturbations for all four models along the vertical axis (x = 500 m) after 700 s for resolutions: (a) 20, (b) 10, and (c) 5 m. All cases were run using 10th order polynomials.

B.   CASE 2: LINEAR HYDROSTATIC MOUNTAIN

1.   Accuracy and Comparison

The modified source code was run and the numerical solution after 10 hours (colored lines) was plotted with and compared to the analytic solution [1] (black lines) (see Figure 10). Only one resolution was used: 1200 m (in x) and 240 m (in z) (20301 grid points). The resulting data for the vertical and horizontal velocity was then used for the comparison (see Table III). As seen in Figure 10, a steady-state mountain wave did form in the appropriate region closely modeling the analytic solution.

Visually comparing Figure 5 to Figure 10, indicated fairly identical solutions, and to further evaluate the similarities in the x-z and x-σ_z model solutions nine variables were again compared: π'_{max}, π'_{min}, u_{max}, u_{min}, w_{max}, w_{min}, θ'_{max}, θ'_{min}, and CPU time (in seconds). Similar to case 1, both x-z and x-σ_z for set 1 code were run in parallel and started at the same time, and then repeated for set 2. The resulting values can be seen in Table III. All four sets of code converge to similar solutions, with only minor differences overall, and the most predominate differences in the w_{max} and w_{min} between x-z coordinates and x-σ_z coordinates, where x-z had greater magnitude values for w_{min} and x-σ_z had greater magnitude values for w_{max}. As seen in case 1, another major difference between the models is the CPU time, which was consistently slower for the x-σ_z (30.03 % increase in time for set 1 and 15.67 % increase in time for set 2) relative to their x-z counterparts. This difference is again being caused by the additional x-σ_z data being passed between program functions and additional calculations being performed ($O(5 * [N_T] * [N_E] * [M_N]^2 * [8 * (N + 1) + 26])$). For this case, there were 1.8793×10^{15} additional operations.

To further evaluate the performance of the four model runs, the RMSE was determined for each run in relation to the analytic solution (see Table IV) for the four variables: π, u, w, and θ. The analytic solution was interpolated to the model grid using a cubic approximation. Once the domain of interest was defined and the nu-

Table III. Case 2: Linear Hydrostatic Mountain. Comparison of modified and unmodified CG1 and CG2 using 1200 m (in x) and 240 m (in z) resolution after 10 hours.

Model	CG1 x-z	CG1 x-σ_z	CG2 x-z	CG2 x-σ_z
π'_{max}	0.1792×10^{-5}	0.1786×10^{-5}	0.1781×10^{-5}	0.1785×10^{-5}
π'_{min}	-0.1859×10^{-5}	-0.1859×10^{-5}	-0.1840×10^{-5}	-0.1846×10^{-5}
u_{max} (ms^{-1})	0.4072×10^{-1}	0.4070×10^{-1}	0.4034×10^{-1}	0.4044×10^{-1}
u_{min} (ms^{-1})	-0.3506×10^{-1}	-0.3484×10^{-1}	-0.3474×10^{-1}	-0.3473×10^{-1}
w_{max} (ms^{-1})	0.4229×10^{-2}	0.4656×10^{-2}	0.4245×10^{-2}	0.4686×10^{-2}
w_{min} (ms^{-1})	-0.5199×10^{-2}	-0.4279×10^{-2}	-0.5214×10^{-2}	-0.4313×10^{-2}
θ'_{max} (K)	0.2400×10^{-1}	0.2386×10^{-1}	0.2380×10^{-1}	0.2382×10^{-1}
θ'_{min} (K)	-0.3184×10^{-1}	-0.3124×10^{-1}	-0.3157×10^{-1}	-0.3115×10^{-1}
CPU time (s)	0.4519×10^{4}	0.5876×10^{4}	0.6463×10^{4}	0.7476×10^{4}

merical and analytic solutions were on matching grids a bootstrap (random sampling) method was used to calculate the 95 % confidence interval (CI) [11]. The 95% CI was needed in order to determine if the differences observed in the RMSE values were indeed significant. The bootstrap method built the 95% CI by creating a domain (of equal size to the domain of interest) that was populated with random samples from the original numerical and analytic solution pairs, and then the RMSE for the new domain was calculated and stored. This process was iterated 10,000 times, storing the RMSE from each iteration. The derived RMSE values were then sorted and the values at 2.5% and 97.5% of the distribution were taken and compared to the original RMSE in order to establish the 95% CI. The resulting confidence intervals indicate that the differences between the state variable RMS errors are significant, since the ranges do not overlap. For set 1, x-z coordinates exhibited a significantly smaller error for all four variables of interest. For set 2, x-σ_z had lower RMSE values for π and u, but larger RMSE values for w and θ. Taking a closer look at the w RMSE with respect to w_{max} shows that CG1 x-z has a 1.32% error as compared to 4.24%

Table IV. Case 2: Linear Hydrostatic Mountain RMSE. Root-mean-squared errors for the four primary state variables, for both the modified and unmodified codes, after 10 h using 1200 m (in x) and 240 m (in z) resolution and 10th order polynomials.

Model	π	95% Confidence Interval	
CG1 x-z	1.2263×10^{-7}	$+ 8.9354 \times 10^{-10}$	$- 8.9757 \times 10^{-10}$
CG1 x-σ_z	1.2898×10^{-7}	$+ 9.5565 \times 10^{-10}$	$- 9.9149 \times 10^{-10}$
CG2 x-z	1.2121×10^{-7}	$+ 9.1186 \times 10^{-10}$	$- 9.1327 \times 10^{-10}$
CG2 x-σ_z	1.2200×10^{-7}	$+ 9.0255 \times 10^{-10}$	$- 9.1227 \times 10^{-10}$
Model	u (ms^{-1})	95% Confidence Interval	
CG1 x-z	2.2561×10^{-3}	$+ 1.7930 \times 10^{-5}$	$- 1.8157 \times 10^{-5}$
CG1 x-σ_z	2.4089×10^{-3}	$+ 1.9437 \times 10^{-5}$	$- 1.9878 \times 10^{-5}$
CG2 x-z	2.2481×10^{-3}	$+ 1.8488 \times 10^{-5}$	$- 1.8337 \times 10^{-5}$
CG2 x-σ_z	2.2808×10^{-3}	$+ 1.8327 \times 10^{-5}$	$- 1.8985 \times 10^{-5}$
Model	w (ms^{-1})	95% Confidence Interval	
CG1 x-z	6.8468×10^{-5}	$+ 6.6744 \times 10^{-7}$	$- 6.6389 \times 10^{-7}$
CG1 x-σ_z	1.9757×10^{-4}	$+ 2.1074 \times 10^{-6}$	$- 2.1468 \times 10^{-6}$
CG2 x-z	5.1424×10^{-5}	$+ 5.0225 \times 10^{-7}$	$- 5.0915 \times 10^{-7}$
CG2 x-σ_z	1.9311×10^{-4}	$+ 2.0131 \times 10^{-6}$	$- 2.0667 \times 10^{-6}$
Model	θ (K)	95% Confidence Interval	
CG1 x-z	1.6565×10^{-3}	$+ 1.3026 \times 10^{-5}$	$- 1.3132 \times 10^{-5}$
CG1 x-σ_z	1.8506×10^{-3}	$+1.4243 \times 10^{-5}$	$- 1.4748 \times 10^{-5}$
CG2 x-z	1.6360×10^{-3}	$+ 1.4812 \times 10^{-5}$	$- 1.4410 \times 10^{-5}$
CG2 x-σ_z	1.7208×10^{-3}	$+ 1.4790 \times 10^{-5}$	$- 1.5027 \times 10^{-5}$

for CG1 x-σ_z and that CG2 x-z has a 0.99% error as compared to 4.12% for CG2 x-σ_z, which is substantial for both sets.

In order to verify that a steady-state solution for the linear hydrostatic mountain case was achieved, the momentum flux was derived using:

$$m(z) = \int_{-\infty}^{\infty} \bar{\rho}(z) u(x,z) w(x,z) dx \qquad (5.1)$$

where $\bar{\rho}(z)$ is the reference density as a function of height (z). [12] Additionally, the analytic hydrostatic momentum flux (from linear theory) is given by:

$$m^H(z) = -\frac{\pi_c}{4}\bar{\rho}_s\bar{u}_s\mathcal{N}h_c^2 \tag{5.2}$$

where m^H denotes the analytic hydrostatic momentum flux, $\bar{\rho}_s$ is the reference density at the surface, $\bar{u}_s$ is the horizontal velocity values at the surface, $\mathcal{N}$ is the Brunt-Väisälä frequency, and h_c is the height of the mountain. [1] The momentum flux was then normalized by $m(z)/m^H(z)$, and will be the value discussed in this thesis. In Figure 11, the normalized momentum flux for all four model runs, using 1200 m (in x) and 240 m (in z) resolution, was plotted for 2 h, 4 h, 6 h, 8 h, and 10 h. As seen in Figure 11, all four model runs converge to a steady state solution after 10 h. Of the four model simulations, the CG1 x-z and CG2 x-z models yielded results that were far better than the x-σ_z models, with values between 0.95 and 1.01 versus between 0.65 and 1.15 as seen for the x-σ_z runs.

2.   Conclusions

All four model runs for the linear hydrostatic mountain case did develop a steady-state mountain wave over a single peak indicating that the model dynamics are resolving the scenario accurately. Additionally, the models converged to nearly identical solutions. Overall, for both set 1 and set 2 of the Navier-Stokes equations, the x-σ_z coordinates performed slightly worse than their x-z counterparts (reflected in the RMSE, the normalized momentum flux, and most pronounced in w with an approximately four times larger RMSE with respect to w_{max}) and as seen in the previous case did show a noticeable increase in computational expense. These results appear to indicate that when the model dynamics are resolved using purely explicit time integration methods that there is no clear advantage in using x-σ_z over x-z and will be further explored into the next case.

i)

ii)

Figure 10. Case 2: Linear Hydrostatic Mountain. Resulting horizontal velocity (a) and vertical velocity (b) using modified (i) CG1 and (ii) CG2 source codes [1] after 10 h for the resolution of 1200 m (in x) and 240 m (in z) (colored lines) plotted with the analytic solution (black lines). All cases were run using 10th order polynomials, with contours from -0.025 to 0.025 ms^{-1} with an interval of 0.005 ms^{-1}, (a), and -0.005 to 0.005 ms^{-1} with an interval of 0.0005 ms^{-1}, (b).
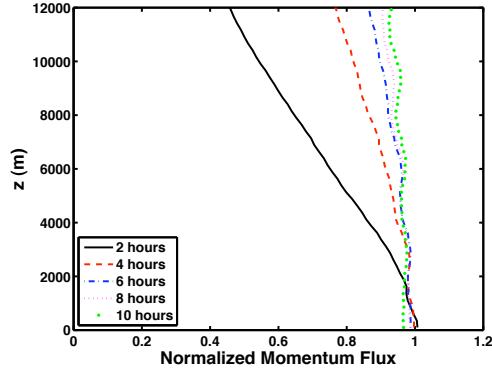
i)
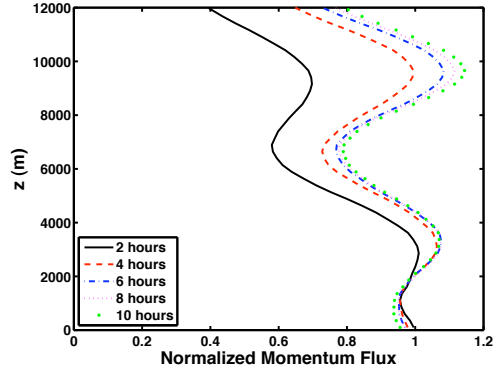
ii)
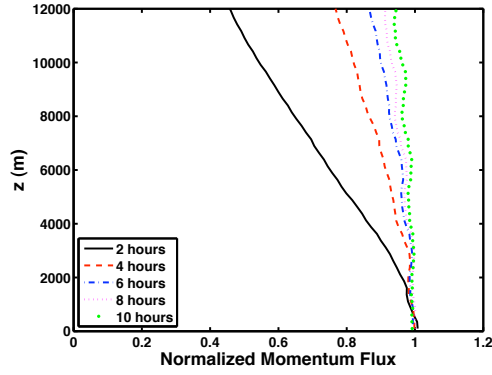
iii)
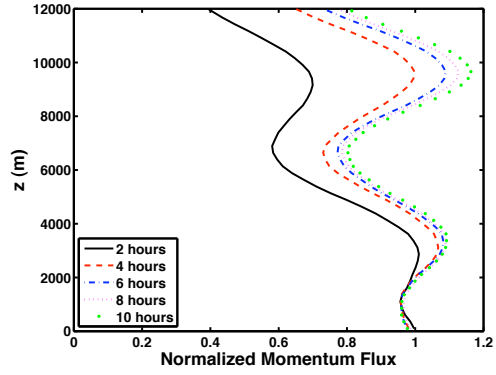
iv)

Figure 11. Case 2: Linear Hydrostatic Mountain. Normalized momentum flux for the resolution of 1200 m (in x) and 240 m (in z), at times 2 h, 4 h, 6 h, 8 h, and 10 h for the four model runs: (i) CG1 x-z, (ii) CG1 x-σ_z, (iii) CG2 x-z, and (iv) CG2 x-σ_z.

C.   CASE 3: LINEAR NON-HYDROSTATIC MOUNTAIN

1.   Accuracy and Comparison

For the final case, both CG1 x-σ_z and CG2 x-σ_z were run and the numerical solution after 5 hours (colored lines) was plotted and compared to the analytic solution [1] (black lines) (see Figure 12). Similar to case 2, only one resolution was used: 360 m (in x) and 300 m (in z) (40501 grid points). The resulting data for the vertical and horizontal velocity was then used for the comparison (see Table V). As seen in Figure 12, a steady-state mountain wave did form in the appropriate region, there were some apparent contrasts with the analytic solution for CG1 x-σ_z and CG2 x-σ_z.

Visually comparing Figure 12.i.a,b to Figure 12.ii.a,b indicated fairly similar solutions for the horizontal velocities (a). Comparing the vertical velocities (b) revealed a similar pattern in the placement of the steady state waves between x-z and x-σ_z, but the x-σ_z coordinates appear to have significantly stronger oscillations than the x-z coordinates. To further evaluate the x-z and x-σ_z model solutions nine variables were again compared: π'_{max}, π'_{min}, u_{max}, u_{min}, w_{max}, w_{min}, θ'_{max}, θ'_{min}, and CPU time (in seconds). Similar to case 1 and case 2, both x-z and x-σ_z for set 1 code were run in parallel and started at the same time, and then repeated for set 2. The resulting values can be seen in Table V. All four sets of code converge to steady state solutions, with larger differences than what were observed in case 2, but had only minor differences in π'_{max}, π'_{min}, u_{max}, and u_{min}. The most predominate differences were observed in the w_{max} and w_{min} between x-z coordinates and x-σ_z coordinates, where x-σ_z had greater magnitude values for both state variables. There was also a noticeable difference in θ'_{max} and θ'_{min} between x-z coordinates and x-σ_z coordinates for both sets. Consistent with the previous two cases, the CPU time was slower for both x-σ_z model simulations (52.06 % increase in time for set 1 and 15.12 % increase in time for set 2) relative to the x-z simulations. This difference is again being caused

by the additional x-σ_z data being passed between program functions and additional calculations being performed ($O(5*[N_T]*[N_E]*[M_N]^2*[8*(N+1)+26])$). For this case, there were 1.8700×10^{16} additional operations.
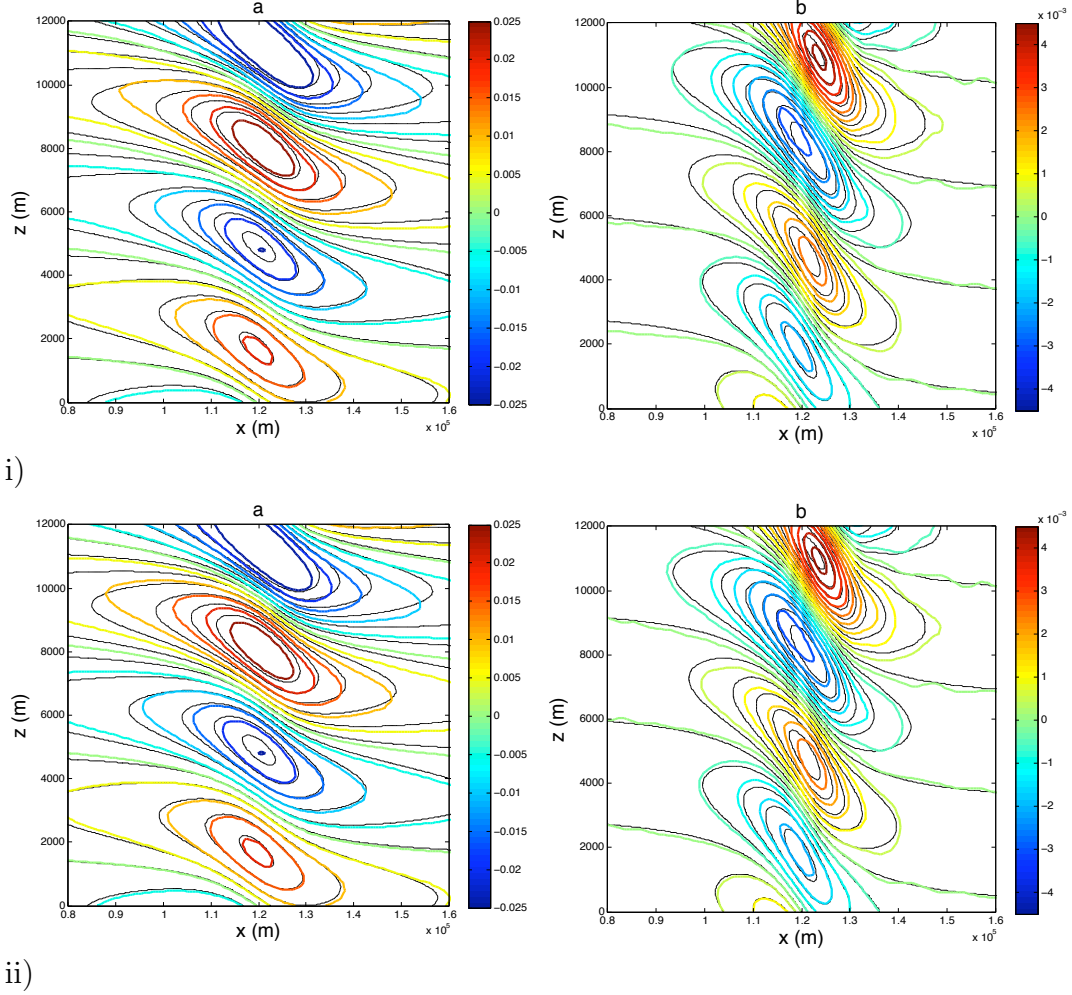

i)

ii)
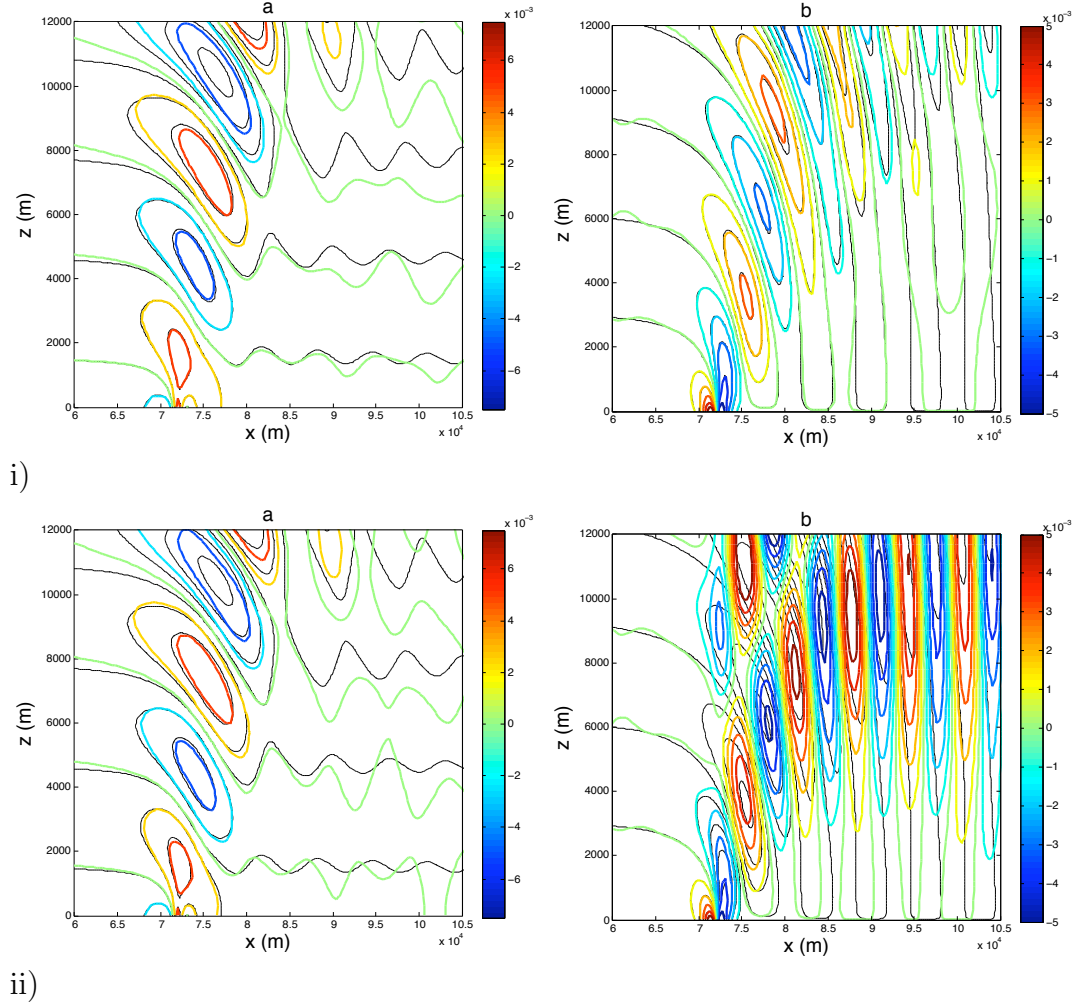
Figure 12. Case 3: Linear Non-Hydrostatic Mountain. Resulting horizontal velocity (a) and vertical velocity (b) using both modified, (ii), and unmodified, (i), CG1 source codes [1] after 5 h for the resolution of 360 m (in x) and 300 m (in z) (colored lines) plotted with the analytic solution (black lines). All cases were run using 10th order polynomials, with contours from -0.025 to 0.025 ms^{-1} with an interval of 0.005 ms^{-1}, (a), and -0.005 to 0.005 ms^{-1} with an interval of 0.0005 ms^{-1}, (b).
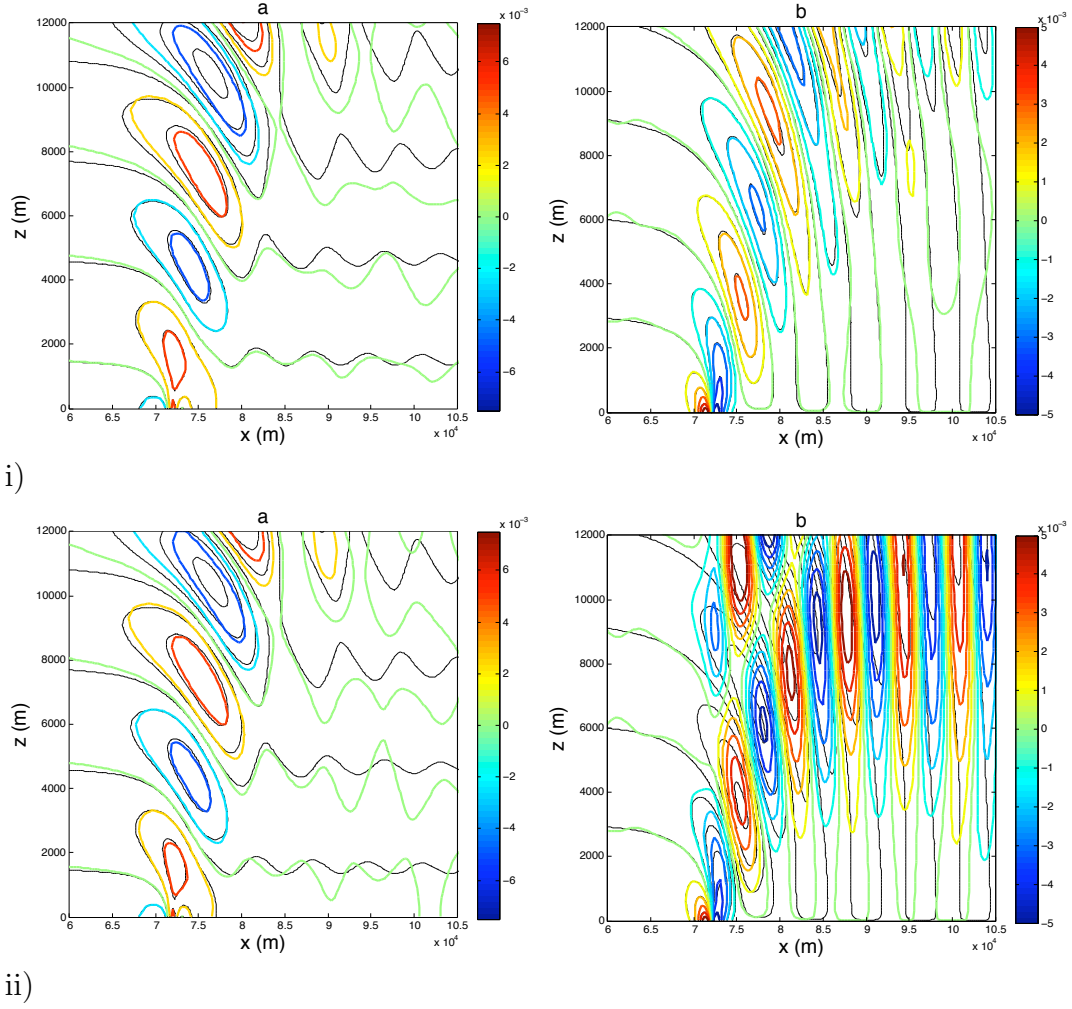
Figure 13. Case 3: Linear Non-Hydrostatic Mountain. Resulting horizontal velocity (a) and vertical velocity (b) using both modified, (ii), and unmodified, (i), CG2 source codes [1] after 5 h for the resolution of 360 m (in x) and 300 m (in z) (colored lines) plotted with the analytic solution (black lines). All cases were run using 10th order polynomials, with contours from -0.025 to 0.025 ms^{-1} with an interval of 0.005 ms^{-1}, (a), and -0.005 to 0.005 ms^{-1} with an interval of 0.0005 ms^{-1}, (b).

To further evaluate the performance of the four model runs, the Root Mean Squared Error (RMSE) was determined for each run in relation to the analytic solution (see Table VI) for the four variables: π, u, w, and θ. Once again, with the domain of interest defined and the numerical and analytic solutions on matching grids a

Table V. Case 3: Linear Non-Hydrostatic Mountain. Comparison of modified and unmodified CG1 and CG2 using 360 m (in x) and 300 m (in z) resolution after 5 h.

Model	CG1 x-z	CG1 x-σ_z	CG2 x-z	CG2 x-σ_z
π'_{max}	0.2281×10^{-6}	0.2059×10^{-6}	0.2276×10^{-6}	0.2045×10^{-6}
π'_{min}	-0.2631×10^{-6}	-0.2628×10^{-6}	-0.2599×10^{-6}	-0.2654×10^{-6}
u_{max} (ms^{-1})	0.8257×10^{-2}	0.8194×10^{-2}	0.8229×10^{-2}	0.8191×10^{-2}
u_{min} (ms^{-1})	-0.7163×10^{-2}	-0.6910×10^{-2}	-0.7120×10^{-2}	-0.6903×10^{-2}
w_{max} (ms^{-1})	0.6035×10^{-2}	0.6696×10^{-2}	0.6035×10^{-2}	0.6694×10^{-2}
w_{min} (ms^{-1})	-0.6037×10^{-2}	-0.7989×10^{-2}	-0.6037×10^{-2}	-0.7988×10^{-2}
θ'_{max} (K)	0.2962×10^{-2}	0.4275×10^{-2}	0.2969×10^{-2}	0.4262×10^{-2}
θ'_{min} (K)	-0.2846×10^{-2}	-0.3987×10^{-2}	-0.2846×10^{-2}	-0.3991×10^{-2}
CPU time (s)	0.6166×10^{4}	0.9376×10^{4}	0.1720×10^{5}	0.1980×10^{5}

bootstrap (random sampling) method, using 10,000 iterations, was used to calculate the 95 % CI [11]. The resulting confidence intervals again indicate that the differences between the state variable RMS errors are significant, since the ranges do not overlap. Both CG1 x-z and CG2 x-z exhibited a smaller error for all four variables of interest, with the most significant difference in RMSE values for w and θ. Similar to the previous case, taking a closer look at the w RMSE with respect to w_{max} shows that CG1 x-z has a 2.17% error as compared to 20.25% for CG1 x-σ_z and that CG2 x-z has a 2.17% error as compared to 20.26% for CG2 x-σ_z, which is even larger than observed in the linear hydrostatic case.

Similar to the previous case, in order to verify that a steady-state solution for the linear non-hydrostatic mountain case was achieved, the momentum flux was again derived using Eq. (5.1). [12] For this case, the analytic non-hydrostatic momentum flux is given by:

$$m^{NH}(z) = -0.457m^{H}(z)$$

where m^{NH} denotes the analytic non-hydrostatic momentum flux and m^{H} denotes

the analytic hydrostatic momentum flux (Eq. (5.2)). [13] The momentum flux was then normalized by $m(z)/m^{NH}(z)$, and will be the value discussed in this thesis. In Figure 14, the normalized momentum flux for all four model runs, using 360 m (in x) and 300 m (in z) resolution, was plotted for 1 h, 2 h, 3 h, 4 h, and 5 h. The resulting momentum flux for this case resembled the same pattern as seen in case 2. Figure 14 indicates that all four models converge to a steady state solution after 5 h. Of the four model simulations, the CG1 x-z and CG2 x-z models yielded results that were far better than the x-σ_z models, with values between 0.95 and 1.0 versus between 0.85 and 1.18 as seen for the x-σ_z runs.

2.   Conclusions

All four model runs for the linear non-hydrostatic mountain case did develop a steady-state mountain wave over a single peak indicating that the model dynamics are resolving the scenario. Additionally, the models converged to nearly identical patterns, but with varying oscillation intensities. Similar to the results seen in case 2, for both set 1 and set 2 of the Navier-Stokes equations, the x-σ_z coordinates performed worse than their x-z counterparts (reflected in the RMSE, the normalized momentum flux , and most pronounced in w with an approximately nine times larger RMSE with respect to w_{max}) and did show a noticeable increase in computational expense. These results further indicate that when the model dynamics are resolved using purely explicit time integration methods that there is no significant benefit in using x-σ_z over x-z and that there is a significant degradation, but the results are promising since the models are converging to a fairly representative steady-state solution. Although the filters and boundary conditions are designed to be independent of z and σ_z, it cannot be ruled out that a special treatment could be explored. With more research into the filters and boundary conditions, the degradation could be made minimal enough that x-σ_z is worth using with semi-implicit methods.

Table VI. Case 3: Linear Non-Hydrostatic Mountain RMSE. Root-mean-squared errors for the four primary state variables, for both the modified and unmodified codes, after 5 h using 360 m (in x) and 300 m (in z) resolution and 10th order polynomials.

| Model	π	95% Confidence Interval	
CG1 x-z	1.5939×10^{-8}	+ 1.0987×10^{-10}	- 1.1296×10^{-10}
CG1 x-σ_z	2.2354×10^{-8}	+ 1.8145×10^{-10}	- 1.8058×10^{-10}
CG2 x-z	1.6391×10^{-8}	+ 1.1292×10^{-10}	- 1.1720×10^{-10}
CG2 x-σ_z	2.2931×10^{-8}	+ 1.8913×10^{-10}	- 1.8253×10^{-10}
Model	u (ms^{-1})	95% Confidence Interval	
CG1 x-z	4.6827×10^{-4}	+ 3.5124×10^{-6}	- 3.5127×10^{-6}
CG1 x-σ_z	5.3962×10^{-4}	+ 4.3611×10^{-6}	- 4.3393×10^{-6}
CG2 x-z	4.8941×10^{-4}	+ 3.6237×10^{-6}	- 3.6915×10^{-6}
CG2 x-σ_z	5.3838×10^{-4}	+ 4.3711×10^{-6}	- 4.3818×10^{-6}
Model	w (ms^{-1})	95% Confidence Interval	
CG1 x-z	1.3125×10^{-4}	+ 1.2912×10^{-6}	- 1.3101×10^{-6}
CG1 x-σ_z	1.6180×10^{-3}	+ 1.2733×10^{-5}	- 1.2822×10^{-5}
CG2 x-z	1.3105×10^{-4}	+ 1.3205×10^{-6}	- 1.3154×10^{-6}
CG2 x-σ_z	1.6180×10^{-3}	+ 1.2727×10^{-5}	- 1.2691×10^{-5}
Model	θ (K)	95% Confidence Interval	
CG1 x-z	1.6683×10^{-4}	+ 1.3953×10^{-6}	- 1.4053×10^{-6}
CG1 x-σ_z	5.3367×10^{-4}	+ 4.7779×10^{-6}	- 4.7821×10^{-6}
CG2 x-z	1.7285×10^{-4}	+ 1.4982×10^{-6}	- 1.4429×10^{-6}
CG2 x-σ_z	5.3364×10^{-4}	+ 4.7514×10^{-6}	- 4.7686×10^{-6}
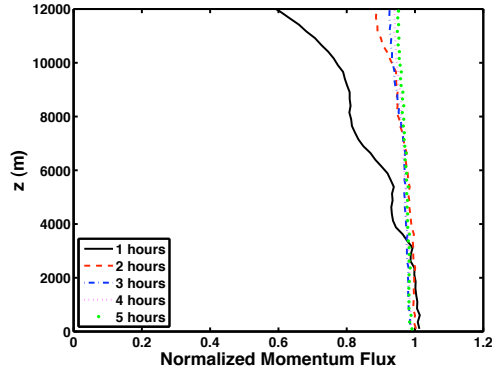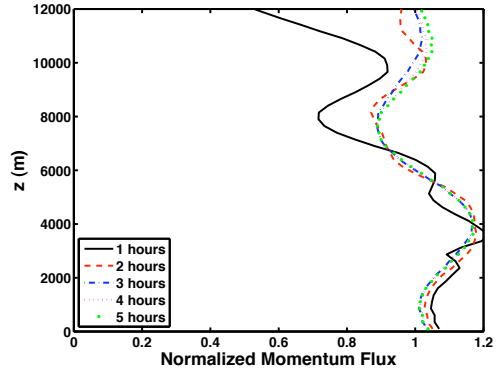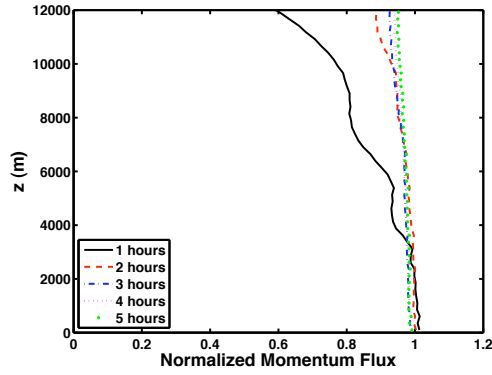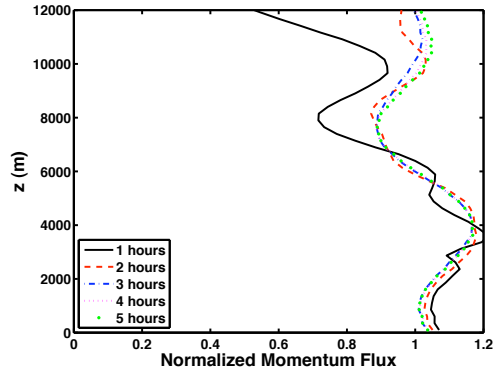
Figure 14. Case 3: Linear Non-Hydrostatic Mountain. Normalized momentum flux for the resolution of 360 m (in x) and 300 m (in z), at times 1 h, 2 h, 3 h, 4 h, and 5 h for the four model runs: (i) CG1 x-z, (ii) CG1 x-σ_z, (iii) CG2 x-z, and (iv) CG2 x-σ_z.

THIS PAGE INTENTIONALLY LEFT BLANK

VI.   CONCLUSIONS AND RECOMMENDATIONS

The results from the three test cases yielded promising results. The first of which was that the x-σ_z coordinates functioned properly and did in fact reduce to x-z coordinates in the absence of terrain, as demonstrated in case 1. With the introduction of terrain, all four models converged to nearly identical steady state patterns for both case 2 and case 3 respectively, but had varying oscillation intensities between coordinate systems. Additionally, both case 2 and case 3 x-σ_z model runs did develop a steady-state mountain wave over a single peak (reflected in the normalized momentum flux) indicating that the model dynamics are resolving the scenarios for a linear hydrostatic mountain and a linear non-hydrostatic mountain. In general, the CG1 x-σ_z and CG2 x-σ_z models performed worse than their x-z counterparts. The x-σ_z models had higher RMS errors (as large as nine times greater RMSE values with respect to associated maximum vertical velocities), which were observed predominantly in intensity values and not in placement of steady state features. All three cases for x-σ_z also showed a noticeable increase in computational expense, due to the additional calculations and variables required by the coordinate transformation. These results indicate that though x-σ_z coordinates are not as accurate or efficient as x-z and that there is a significant degradation, with further fine-tuning of the model environment these issues could be made minimal enough that it justifies their use in semi-implicit methods, especially in the vertical, as is already done by various operational mesoscales models.

Past research has demonstrated the utility of using Continuous Galerkin methods, in a x-z framework, for resolving computational fluid dynamics using both fully explicit and semi-implicit time integration methods. This thesis brought Giraldo's 2-D (x-z slice) mesoscale Non-Hydrostatic model one step closer to evaluating and exploiting the full strength of x-σ_z coordinates by transforming the Navier-Stokes

Equations and testing their impacts using fully explicit time integration. Now that the x-σ_z models are functional, the next stage of development will be to implement a semi-implicit time integration method in the vertical (1-D) and compare the results against the 2-D semi-implicit x-z model. Without the work done in this project to transform the x-z equations to x-σ_z, one could not construct semi-implicit methods in the vertical since in x-z, the terrain is coupled to box coordinates and so the coordinates cannot be decoupled. Additionally, the mathematically machinery outlined in this thesis can be used to transform any equation set to any other coordinate system.

The value of this study is far reaching in determining the usefulness of applying a specific coordinate system in the future when developing meteorological and oceanographic models for the U.S. Naval Research Laboratory (NRL) by constituents at the Naval Postgraduate School. In addition, the successful conversion of the non-hydrostatic x-z models to x-σ_z will allow for the straightforward extension of these models to global non-hydrostatic models, since σ will represent the height of the model and x will then represent the spherical shell at each value of σ.

APPENDIX. COEFFICIENTS FOR RK35 METHOD

Coefficient	Value
stage 1:	
α_0	1
β_0	0.377268915331368
stage 2:	
α_0	0
α_1	1
β_0	0.377268915331368
stage 3:	
α_0	0.355909775063327
α_1	0
α_2	0.644090224936674
β_0	0.242995220537396
stage 4:	
α_0	0.367933791638137
α_1	0
α_2	0
α_3	0.632066208361863
β_0	0.238458932846290
stage 5:	
α_0	0
α_1	0
α_2	0.237593836598569
α_3	0
α_4	0.762406163401431
β_0	0.287632146308408

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] F.X. Giraldo and M. Restelli. "A study of continuous and discontinuous Galerkin methods for the NavierStokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases. " *Journal of Computational Physics*, 227:3849–3877, 2008.

[2] R.M. Hodur. "The Naval Research Laboratorys Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS). " *Monthly Weather Review*, 125:1414–1430, 1997.

[3] W.C. Skamarock, J.B. Klemp, J. Dudhia, D.O. Gill, D.M. Baker, W. Wang, and J.G. Powers. A description of the Advanced Research WRF Version 2. NCAR Technical Note NCART/TN-468+STR, 2007.

[4] T. Gal-Chen and C.J. Somerville. "On the use of a coordinate transformation for the solution of the Navier-Stokes equations. " *Journal of Computational Physics*, 17:209–228, 1975.

[5] F.X Giraldo, M. Restelli, and M. Läuter. Semi-Implicit High-Order Methods for the Euler Equations used in Nonhydrostatic Mesoscale Atmospheric Modeling. American Geophysical Union Fall Meeting, 2008.

[6] T.J. De Luca. Performance of Hybrid Eulerian-Lagrangian Semi-Implicit Time Integrators for Nonhydrostatic Mesoscale Atmospheric Modeling. MS thesis, Naval Postgraduate School, 2007.

[7] F.X Giraldo. "Mathematical Principles of Galerkin Methods," class notes for MA 4245, Department of Applied Math, Naval Postgraduate School, Spring 2008.

[8] R.J. Spiteri and S.J. Ruuth. "A new class of optimal high-order strong-stability-preserving time discretization methods. " *SIAM Journal of Numerical Analysis*, 40:469–491, 2002.

[9] D.R. Durran and J.B. Klemp. "A compressible model for the simulation of moist mountain waves. " *Monthly Weather Review*, 111:2341–2361, 1983.

[10] J. Dea, F.X. Giraldo, and B. Neta. High-order Higdon non-reflecting boundary conditions for the linearized Euler equations. NPS Technical Report NPS-MA-07-001, Naval Postgraduate School, Monterey, CA, 2007.

[11] F.A. Eckel. "Verification of Ensemble Forecasts," class notes for MR 4324, Department of Meteorology, Naval Postgraduate School, Summer 2008.

[12] R.B. Smith. "The influence of mountains on the atmosphere. " *Advances in Geophysics*, 21:87, 1979.

[13] J.B. Klemp and D.R. Durran. "An upper boundary condition permitting internal gravity wave radiation in numerical mesoscale models. " *Monthly Weather Review*, 111:430–444, 1983.

[14] R. Benoit, M. Desgagné, P. Pellerin, S. Pellerin, Y. Chartier, and S. Desjardins. "The Canadian MC2: A semi-Lagrangian, semi-implicit wideband atmospheric model suited for finescale process studies and simulation. " *Monthly Weather Review*, 125:2382–2415, 1997.

[15] L. Bonaventura. "A semi-implicit, semi-Lagrangian scheme using the height coordinate for a nonhydrostatic and fully elastic model of atmospheric flows. " *Journal of Computational Physics*, 158:186–213, 2000.

[16] J. Boyd. The erfc-log filter and the asymptotics of the Euler and Vandeven sequence accelerations. in A. V. Ilin and L. R. Scott (eds), Proceedings of the Third International Conference on Spectral and High Order Methods. Houston Journal of Mathematics, Houston, Texas, 1996.

[17] R.L. Carpenter, K.K. Droegemeier, P.R. Woodward, and C.E. Hane. "Application of the piecewise parabolic method (PPM) to meteorological modeling. " *Monthly Weather Review*, 118:586–612, 1990.

[18] B. Cockburn and C-W. Shu. "Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. " *Journal of Scientific Computing*, 16:173–261, 2001.

[19] A. Gassmann. "An improved two-time-level split-explicit integration scheme for non-hydrostatic compressible models. " *Meteorology and Atmospheric Physics*, 88:23–38, 2005.

[20] F.X. Giraldo. A space marching adaptive remeshing technique applied to the 3d euler equations for supersonic flow. PhD dissertation, University of Virginia, 1995.

[21] F.X. Giraldo and T.E. Rosmond. "A scalable spectral element Eulerian atmospheric model (SEE-AM) for NWP: dynamical core tests. " *Monthly Weather Review*, 132:133–153, 2004.

[22] F.X. Giraldo. "Semi-implicit time-integrators for a scalable spectral element atmospheric model. " *Quarterly Journal of the Royal Meteorological Society*, 131:2431–2454, 2005.

[23] G. Grell, J. Dudhia, and D. Stauffer. A description of the fifth-generation Penn State/NCAR Mesoscale Model (MM5). NCAR Technical Note NCART/TN-398+STR, 1995.

[24] Z. Janjic. "A nonhydrostatic model based on a new approach. " *Meteorology and Atmospheric Physics.*, 82:271–285, 2003.

[25] J.B. Klemp and D.K. Lilly. "Numerical simulation of hydrostatic mountain waves. " *Journal of the Atmospheric Sciences*, 35:78–107, 1978.

[26] J.B. Klemp and R.B. Wilhelmson. "The simulation of three-dimensional convective storm dynamics. " *Journal of the Atmospheric Sciences*, 35:1070–1096, 1978.

[27] J.P. Pinty, R. Benoit, E. Richard, and R. Laprise. "Simple tests of a semi-implicit semi-Lagrangian model on 2D mountain wave problems. " *Monthly Weather Review*, 123:3042–3058, 1995.

[28] M. Restelli and F.X. Giraldo. "A conservative semi-implicit discontinuous Galerkin method for the Navier-Stokes equations in nonhydrostatic mesoscale atmospheric modeling. " *SIAM Journal on Scientific Computing*, 2007.

[29] A. Robert. "Bubble convection experiments with a semi-implicit formulation of the Euler equations. " *Journal of the Atmospheric Sciences*, 50:1865–1873, 1993.

[30] C. Schar, D. Leuenberger, O. Fuhrer, D. Luthic, and C. Girard. "A new terrain-following vertical coordinate formulation for atmospheric prediction models. " *Monthly Weather Review*, 130:2459–2480, 2002.

[31] W.C. Skamarock and J.B. Klemp. "Efficiency and accuracy of the Klemp-Wilhelmson time-splitting technique. " *Monthly Weather Review*, 122:2623–2630, 1994.

[32] W.C. Skamarock, J.D. Doyle, P. Clark, and N. Wood. A standard test set for nonhydrostatic dynamical cores of NWP models. AMS NWP-WAF Conference Poster P2.17, 2004.

[33] J.M. Straka, R.B. Wilhelmson, L.J. Wicker, J.R. Anderson, and K.K. Droegemeier. "Numerical solutions of a non-linear density current: a benchmark solution and comparisons. " *International Journal for Numerical Methods in Fluids*, 17:1–22, 1993.

[34] L. Wicker and W.C. Skamarock. "A time-splitting scheme for the elastic equations incorporating second-order Runge-Kutta time differencing. " *Monthly Weather Review*, 126:1992–1999, 1998.

[35] L. Wicker and W.C. Skamarock. "Time-splitting methods for elastic models using forward time schemes. " *Monthly Weather Review*, 130:2088–2097, 2002.

[36] M. Xue, K.K. Droegemeier, and V. Wong. "The advanced regional prediction system (ARPS) - a multi-scale nonhydrostatic atmospheric simulation and prediction model. Part I: Model dynamics and verification. " *Meteorology and Atmospheric Physics*, 75:161–193, 2000.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. AF Weather Technical Library
   151 Patton Ave Rm 120
   Asheville, North Carolina 28801-5002

4. U.S. Naval Research Labs
   Monterey, California

5. F.X. Giraldo
   Department of Applied Mathematics
   Naval Postgraduate School
   Monterey, California

6. Maj. Tony Eckel
   Department of Meteorology
   Naval Postgraduate School
   Monterey, California